Messenger Application
+ Project Management System

Group Members

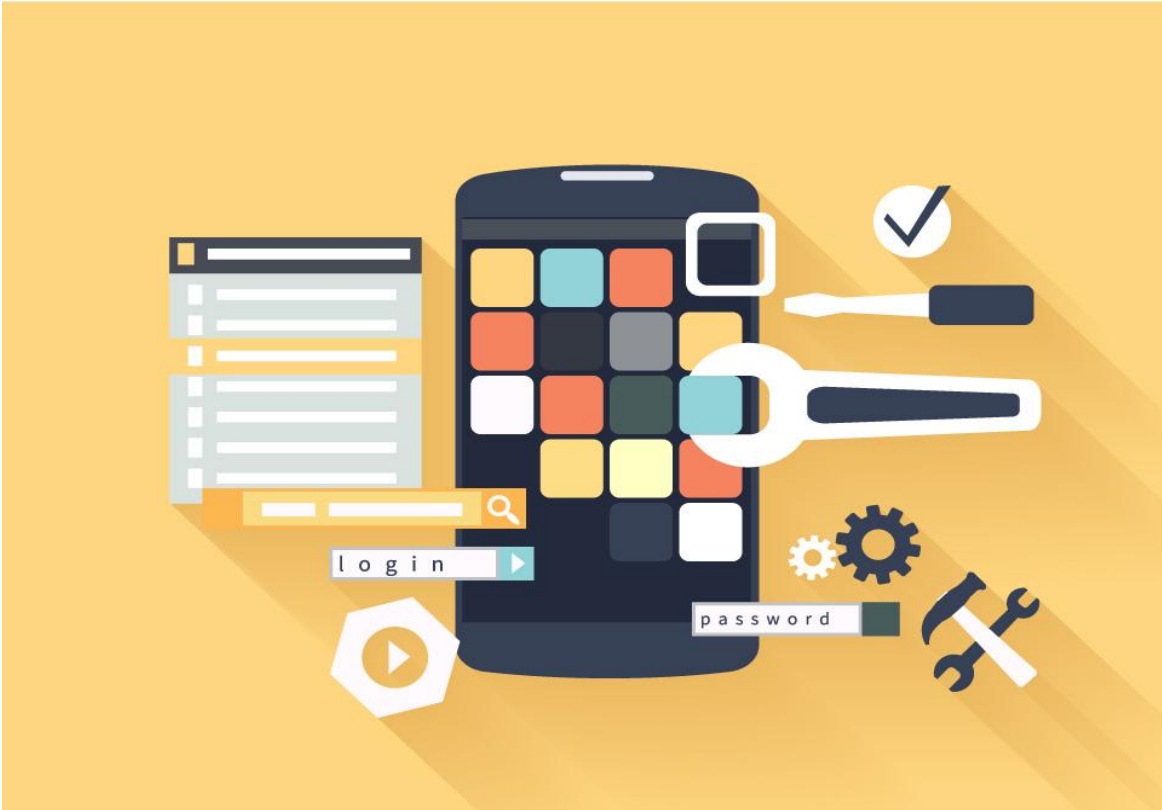
amirphl@bisphone.com

abdollahpourmm@gmail.com

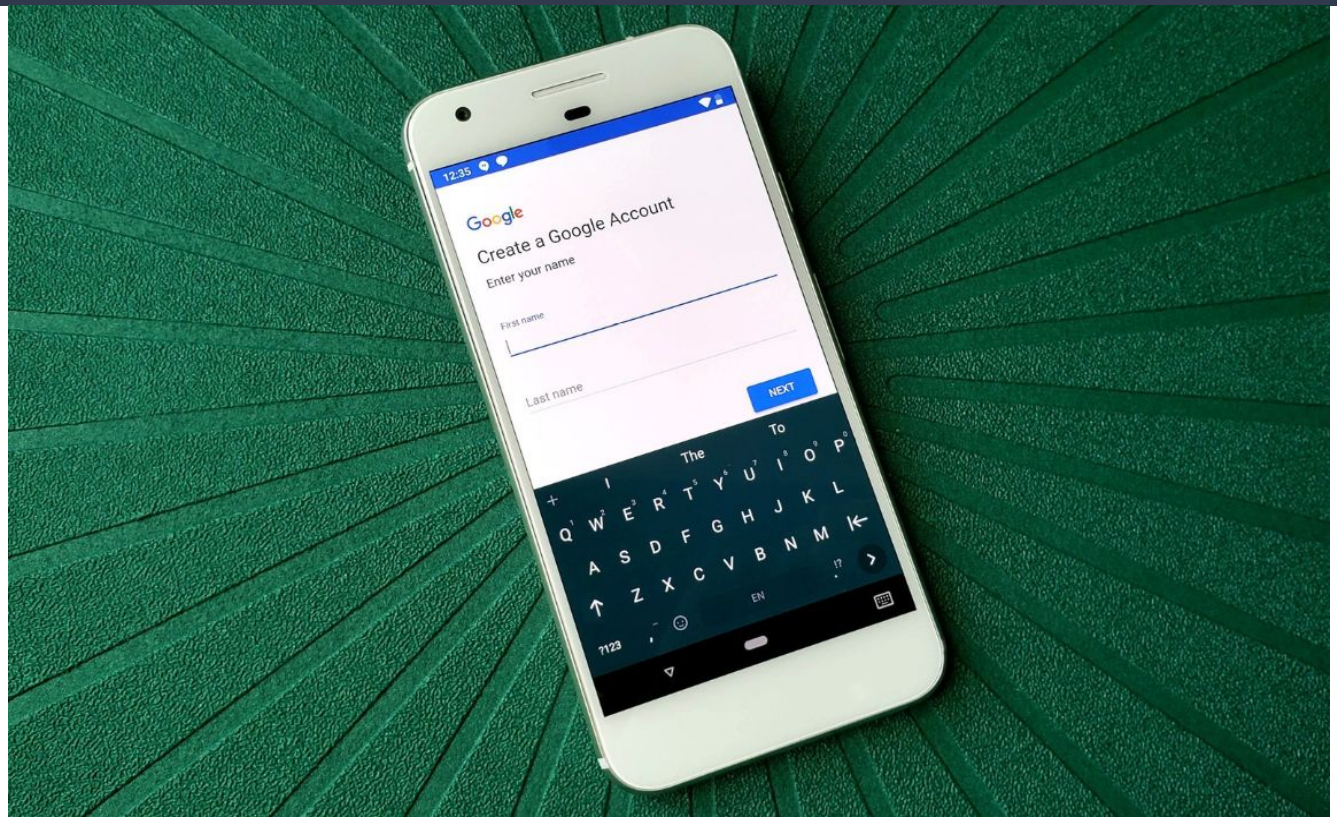
kamyarkh76@gmail.com

sinatr7777777@gmail.com

User Requirements(functional)

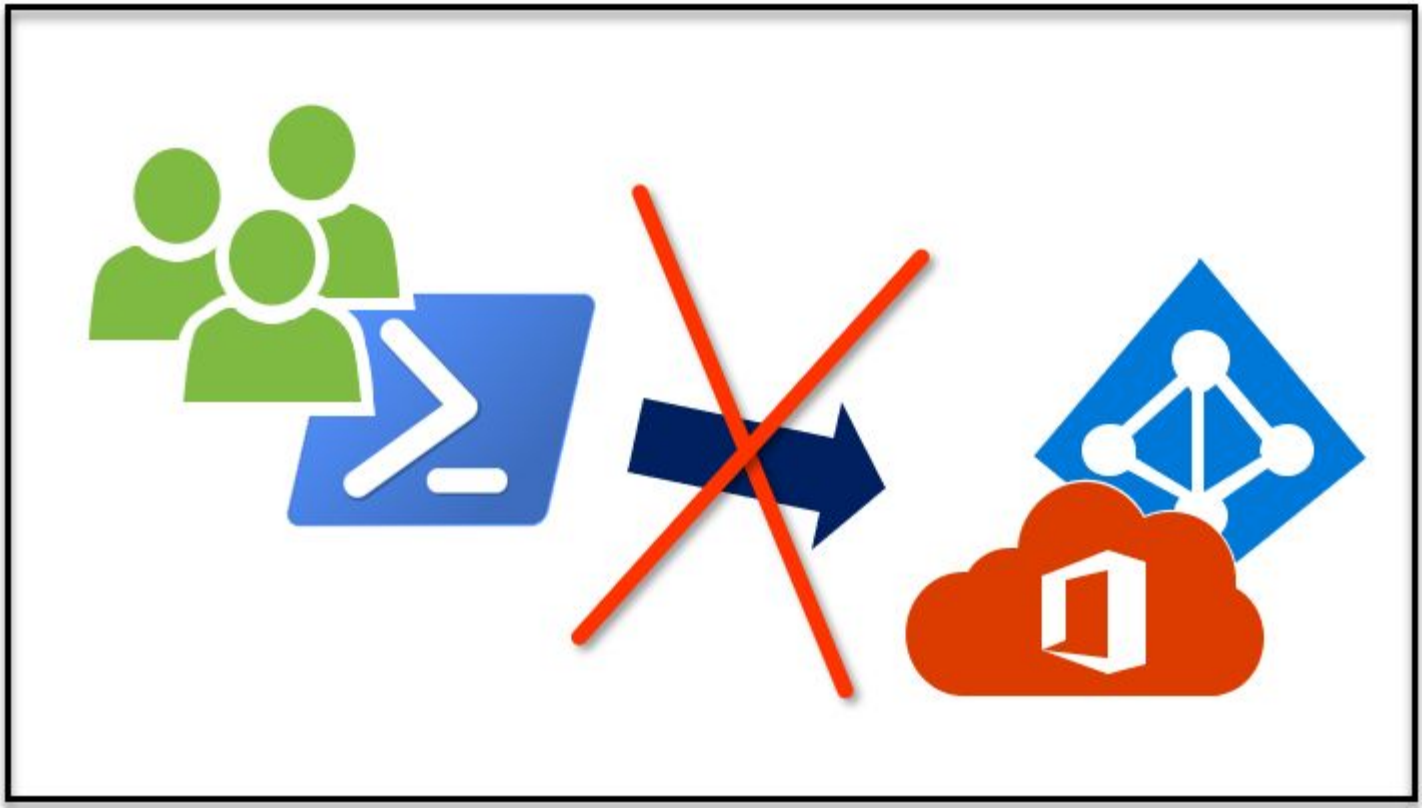


1,2)create account and login and logout

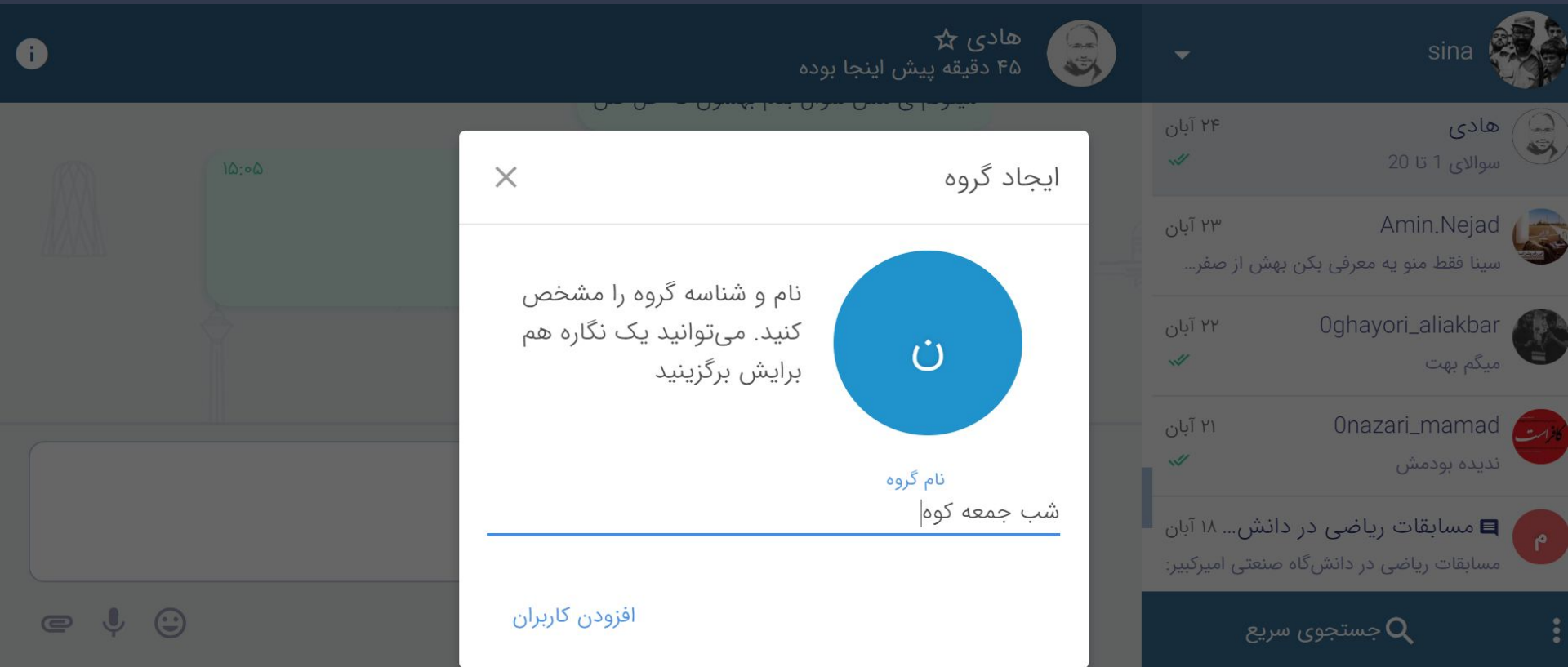


3)send message,media

4)Block user



5,6)Create Group and Team



7,8,9,10)Create Project,Task and Add and remove Member

Create a Project

Project name* Teams in Space

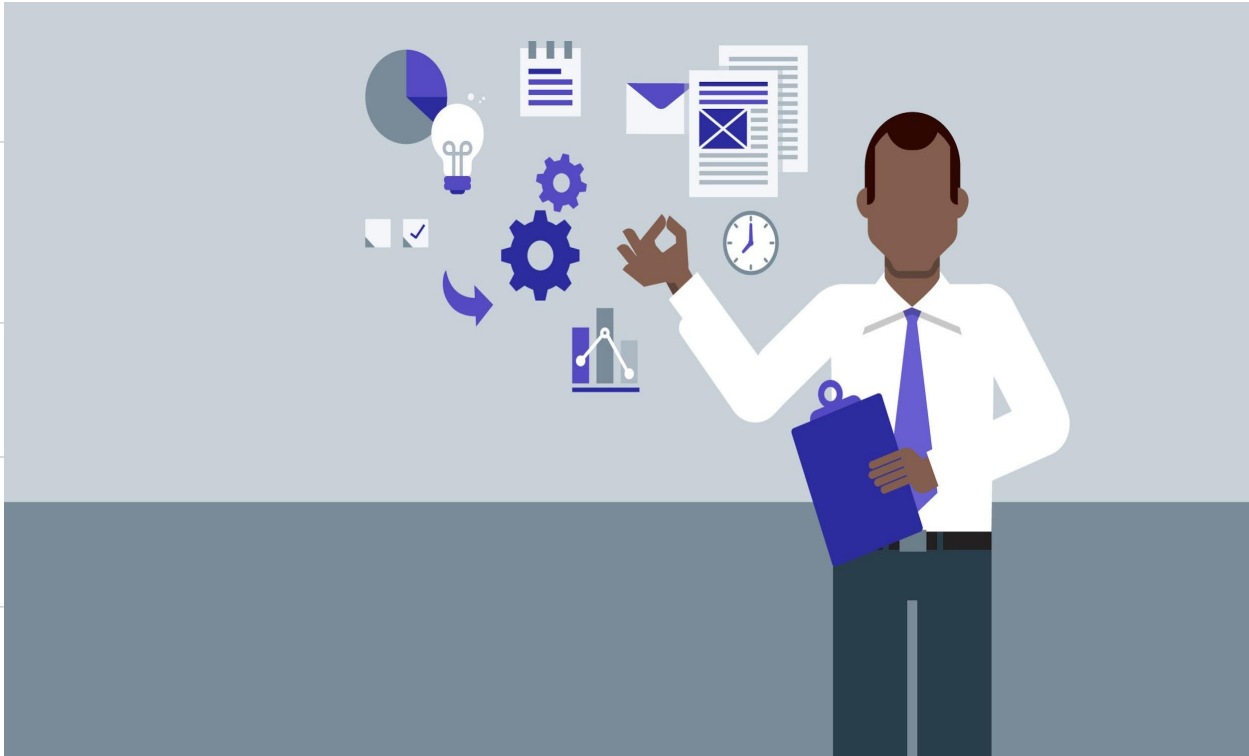
Project key* TIS

Eg. AT (for a project named Atlassian)

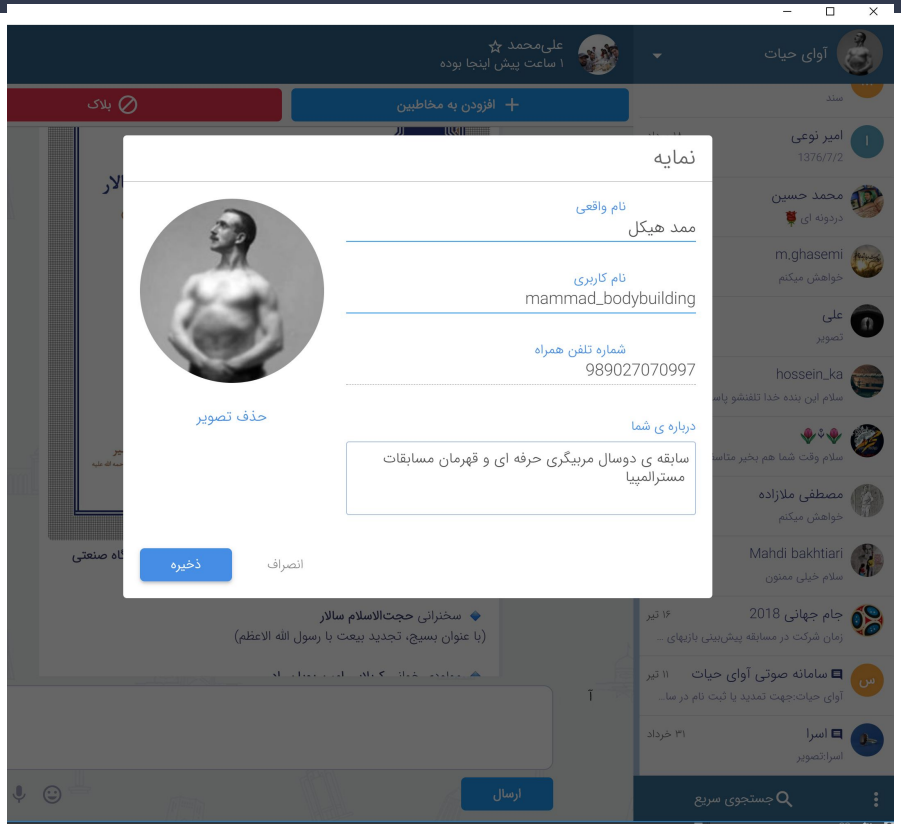
Description Advancing humanity into the cosmos!

Project Avatar  Change avatar

Create project Cancel

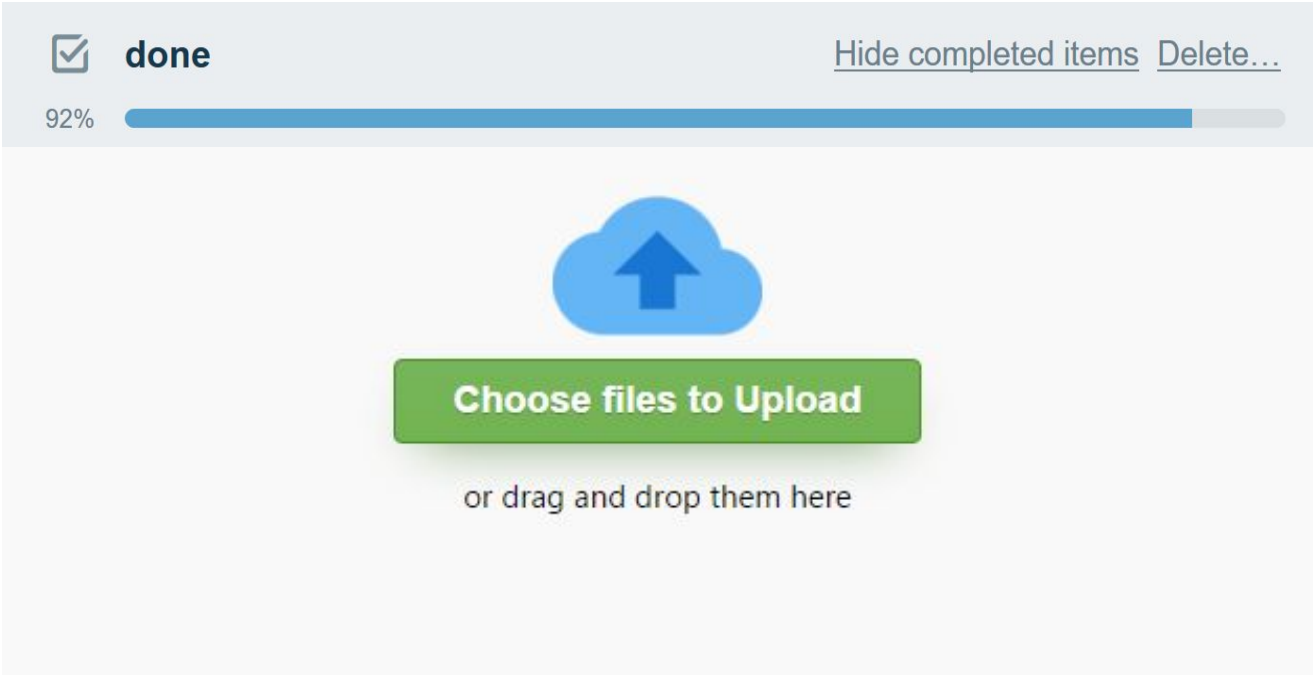


11) user can build his personal profile and personalize user interface

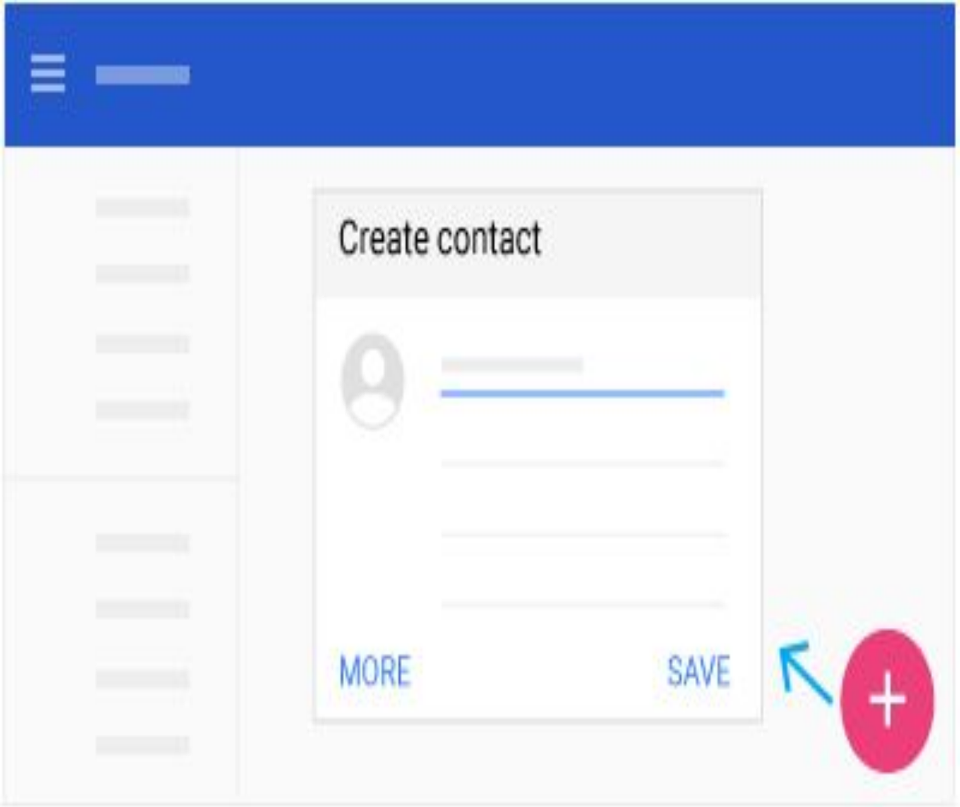


12) grant access by admin (make someone co leader)

13,14) user can create task show his task progress between 0 and 100 and can upload result of task as a file



15,16,17)add,list & update contact



Contact List

Name   Adam Smith Carton

Contact Category	First Name	Last Name	Business/Company	Personal Contact
Business	Shyla Josuf	Kitty	AJK Ltd	(000)-111-2222
Personal	Smith Albert	Philips	Jaguar Insurance	(111)-111-1111
Business	Shyla Josuf	Kitty	AJK Ltd	(000)-111-2222
Personal	Smith Albert	Philips	Jaguar Insurance	(111)-111-1111
Business	Shyla Josuf	Kitty	AJK Ltd	(000)-111-2222
Personal	Smith Albert	Philips	Jaguar Insurance	(111)-111-1111
Business	Shyla Josuf	Kitty	AJK Ltd	(000)-111-2222
Personal	Smith Albert	Philips	Jaguar Insurance	(111)-111-1111
Business	Shyla Josuf	Kitty	AJK Ltd	(000)-111-2222
Personal	Smith Albert	Philips	Jaguar Insurance	(111)-111-1111
Business	Shyla Josuf	Kitty	AJK Ltd	(000)-111-2222
Personal	Smith Albert	Philips	Jaguar Insurance	(111)-111-1111
Business	Shyla Josuf	Kitty	AJK Ltd	(000)-111-2222
Personal	Smith Albert	Philips	Jaguar Insurance	(111)-111-1111
Business	Shyla Josuf	Kitty	AJK Ltd	(000)-111-2222
Personal	Smith Albert	Philips	Jaguar Insurance	(111)-111-1111
Business	Shyla Josuf	Kitty	AJK Ltd	(000)-111-2222
Personal	Smith Albert	Philips	Jaguar Insurance	(111)-111-1111
Business	Shyla Josuf	Kitty	AJK Ltd	(000)-111-2222
Personal	Smith Albert	Philips	Jaguar Insurance	(111)-111-1111

18)Password recovery

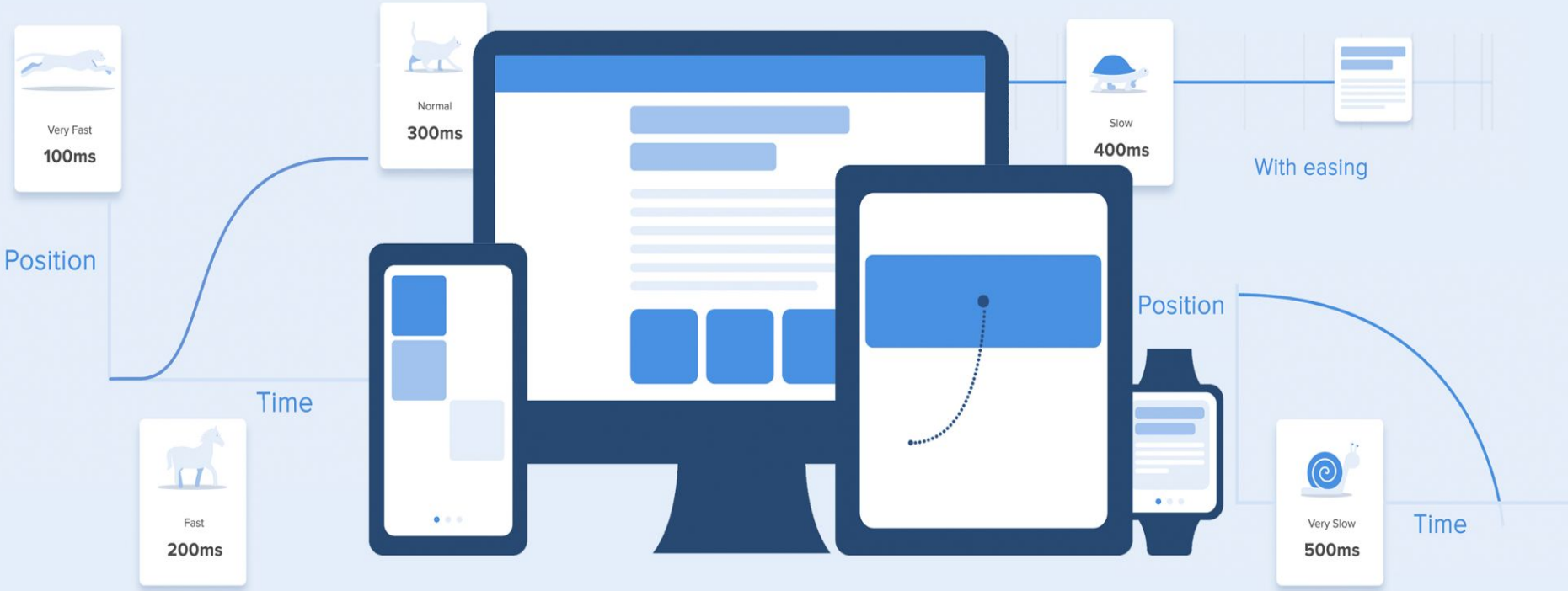
User requirements (non-functional)

1) User can see just information of his team ,group and project

2) Having messages on a disk that can also see messages when they are offline

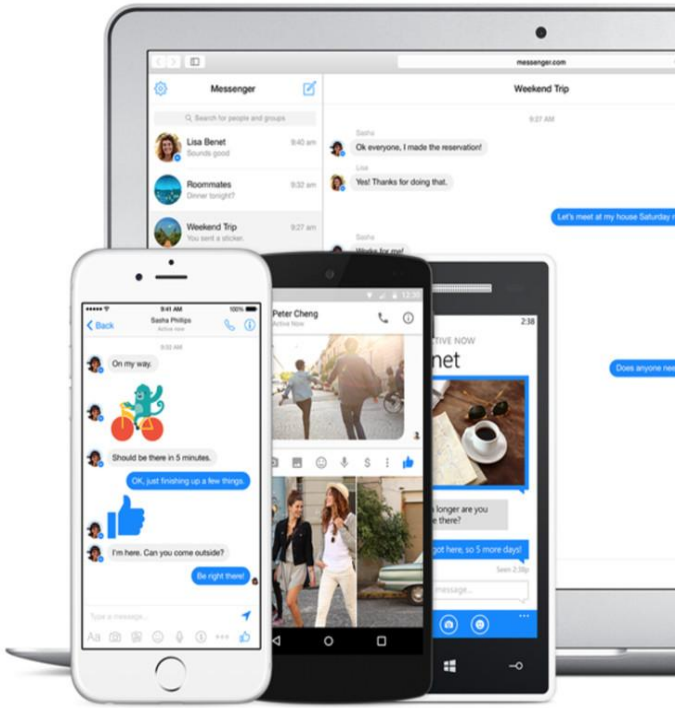
3) User messages must be encrypted and the entire system secured

4)Proper user interface



5)high speed send

6)Web version



Distributed Client–server pattern

The Client–server pattern

In a client–server architecture, the functionality of the system is organized into services, with each service delivered from a separate server. Clients are users of these services and access servers to make use of them.

Why???

This pattern is used when data in a shared database has to be accessed from a range of locations.

Because servers can be replicated, it may also be used when the load on a system is variable.

Our system exactly falls into above criterias.

Client–server pattern

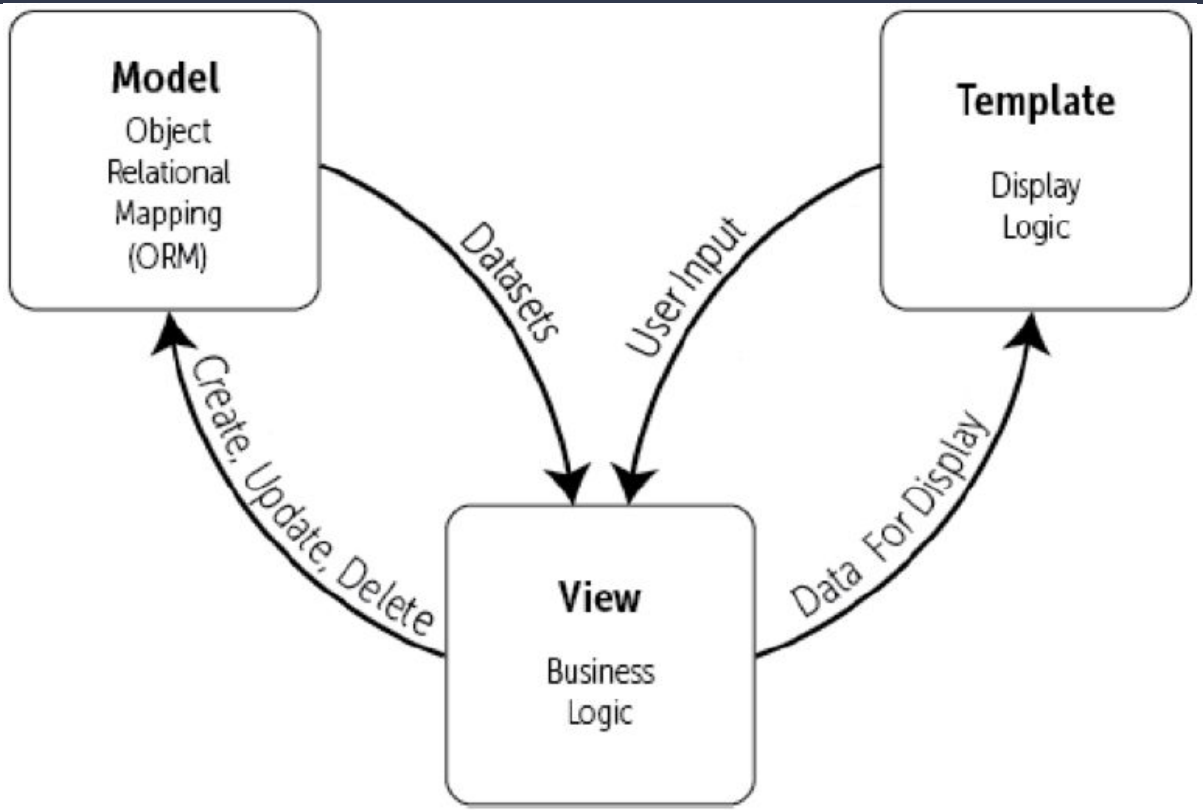
Advantages

A. The principal advantage of this model is that servers can be distributed across a network.
B. General functionality can be available to all clients and does not need to be implemented by all services.
C. SPOF is solved by distributing the servers and assign one of them as NameNode and one as secondary NameNode(redundancy).

Disadvantages

A. Performance may be unpredictable because it depends on the network as well as the system.
B. May be management problems if servers are owned by different organizations

Model-View-Template (MVT)



Model-View-Template (MVT)

The **MVT (Model View Template)** is a software design pattern. It is a collection of three important components Model View and Template. The Model helps to handle database. It is a data access layer which handles the data.

Model: Just like the Model explanation in the MVC pattern , this also takes the same position as the interface or relationship between the data and contains everything related to data access and validation.

Template: This relates to the View in the MVC pattern as it is the presentation layer that handles the presentation logic in the framework and basically controls what should be displayed and how it should be displayed to the user.

View: This part relates to the Controller in the MVC pattern and handles all the business logic that throws down back to the respective templates.It serves as the bridge between the model and the template

The **tiny difference** that can constitute as the most confusing part in all this, is how Django suggests that the View should include the business logic instead of the presentation logic alone as it is in the standard MVC pattern and the Template to take care of all of the presentation logic alone while the MVC pattern does not include a Template component at all. As a result of this, when compared to the standard MVC pattern, Django's design is also referred to as the ***Model-Template-View + Controller*** where Controller is often times omitted because it's already part of the framework.

Model-View-Template (MVT)

Advantages:

Allows the data to change independently of its representation and vice versa. Supports presentation of the same data in different ways with changes made in one representation shown in all of them.
Suitable for web applications.
There are many useful frameworks which are implemented in MVT model and suitable for reuse-oriented approach.

Disadvantages:

Can involve additional code and code complexity when the data model and interactions are simple.

Priority Table

Priority

Aspects of Prioritization:

1 - Importance (by stakeholders)
2 - Cost (The implementation cost is usually estimated by the developing organization)
3 - Impact Rate in Database
4 - Load Amount on network
5 - Educational Needs & Needed Development of backup infrastructure

Priority list

Low (1-6) - This defect can be fixed after the critical ones are fixed.

Medium (7-11) - The defect should be resolved in the subsequent builds.

High (12-15) - The defect must be resolved immediately because the defect is affecting the application to a considerable extent and the relevant modules cannot be used until it's fixed.

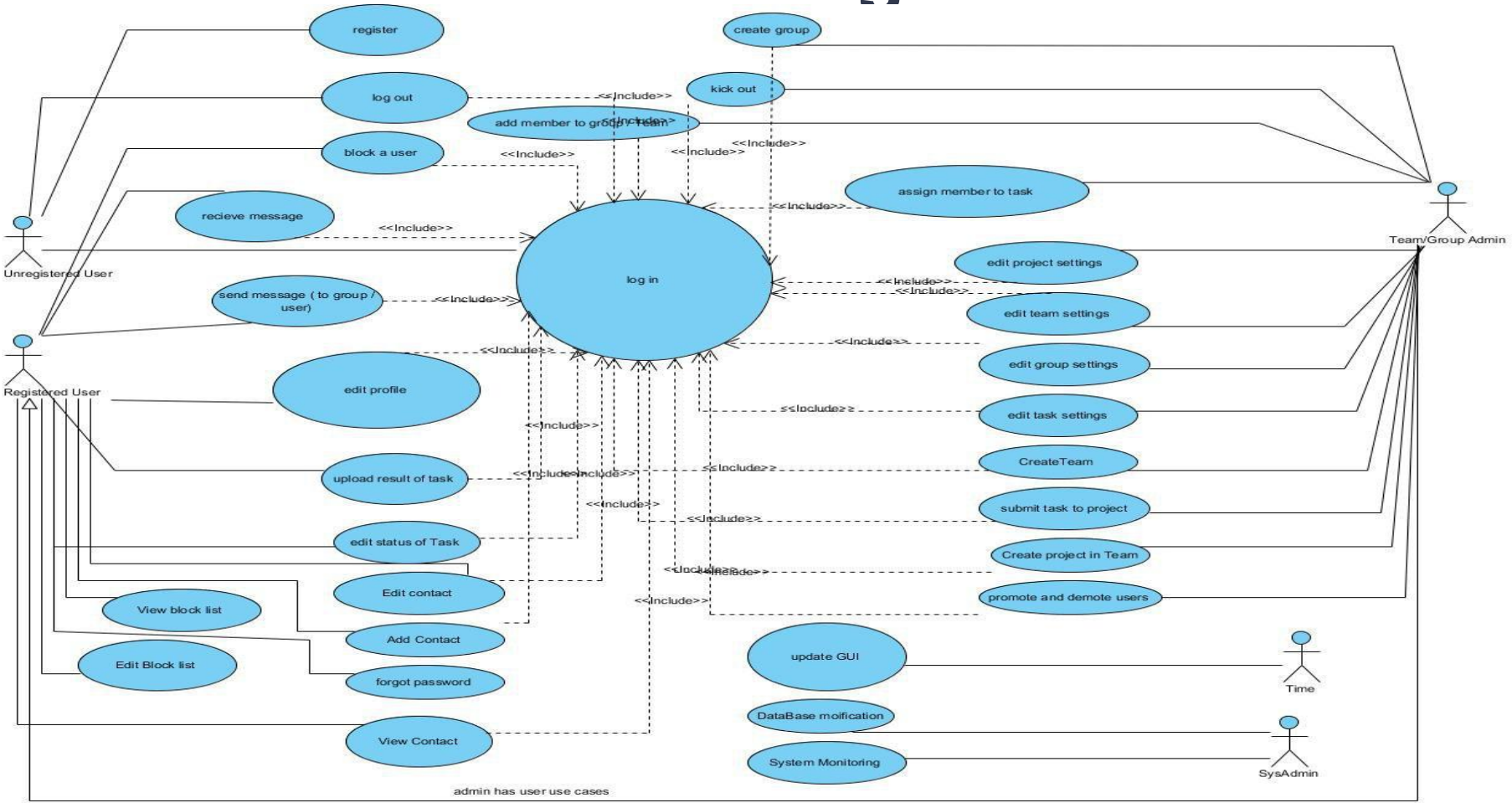
Priority Table

Priority	Total	A5	A4	A3	A2	A1	Task
High	15	3	3	3	3	3	Create account
High	15	3	3	3	3	3	Login & Logout
High	14	2	3	3	3	3	Send message
High	12	3	2	2	3	2	Block user
Medium	11	2	2	2	3	2	Create group
Medium	10	2	2	2	2	2	Create team
Medium	9	1	2	2	2	2	Create project
Medium	8	1	2	2	1	2	Add member
Medium	8	1	2	2	1	2	Kick member

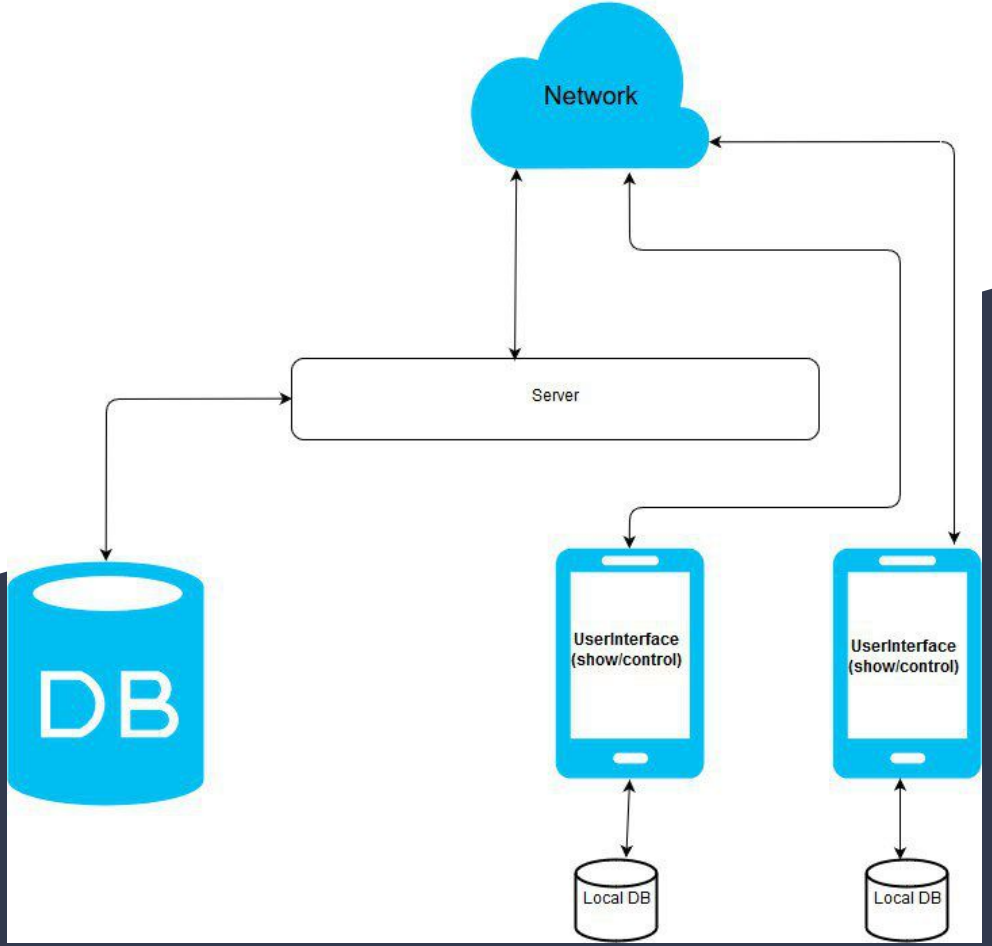
Priority Table

Low	7	2	1	2	1	1	Edit account
Low	7	2	1	1	2	1	Access level
Low	7	1	1	2	1	2	Create task
Low	5	1	1	1	1	1	set task status
Low	7	1	1	2	1	2	Add contact
Low	5	1	1	1	1	1	List of contacts
Low	5	1	1	1	1	1	Edit contact

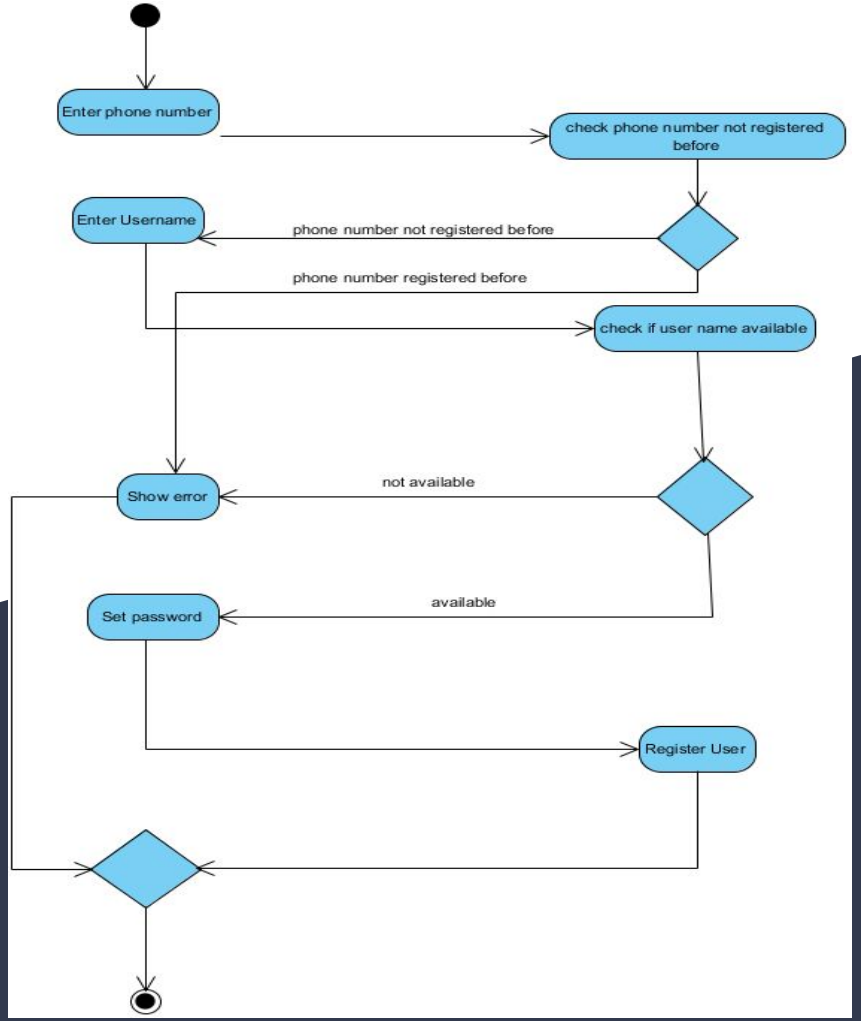
Use Case Diagram



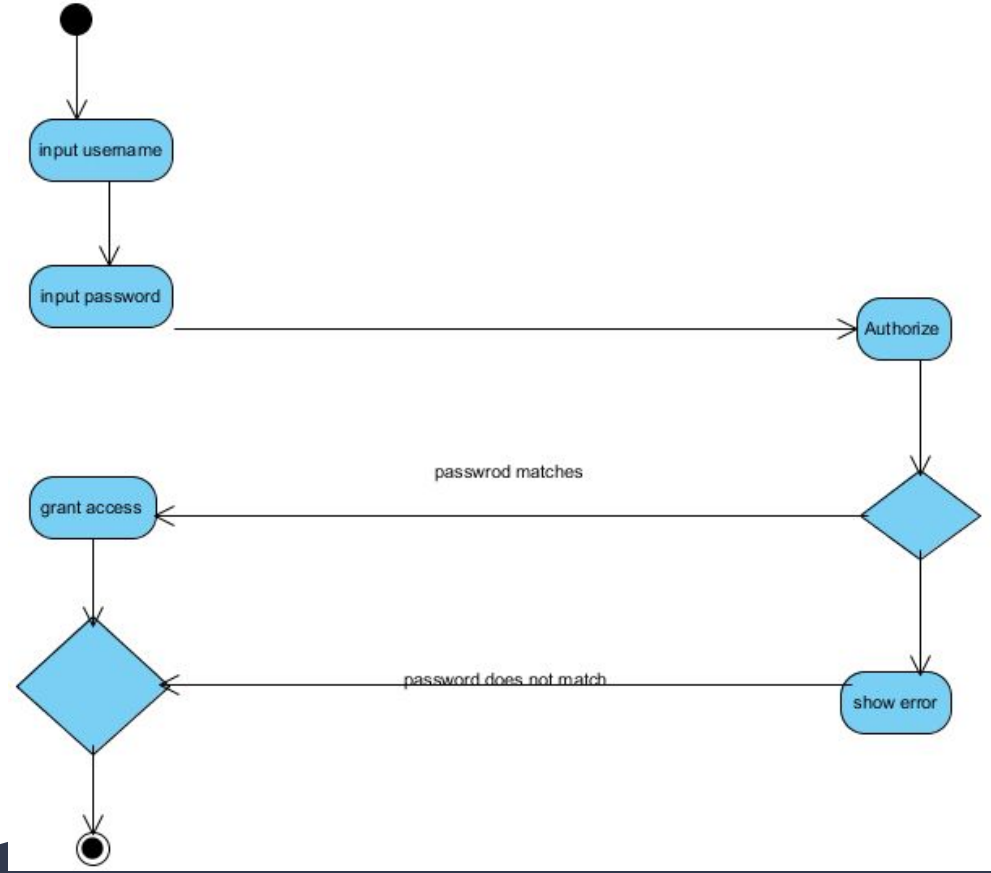
Block Diagram



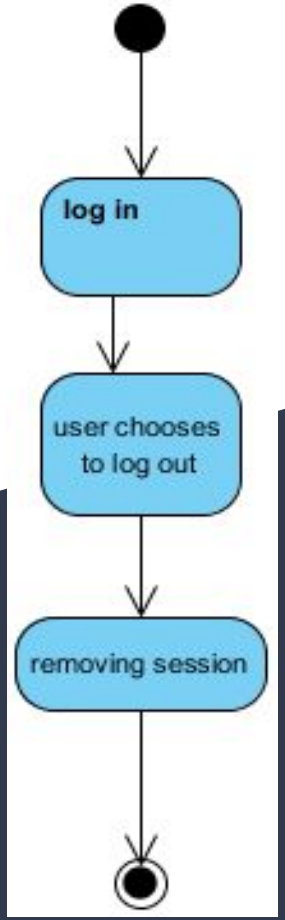
Create Account Activity Diagram



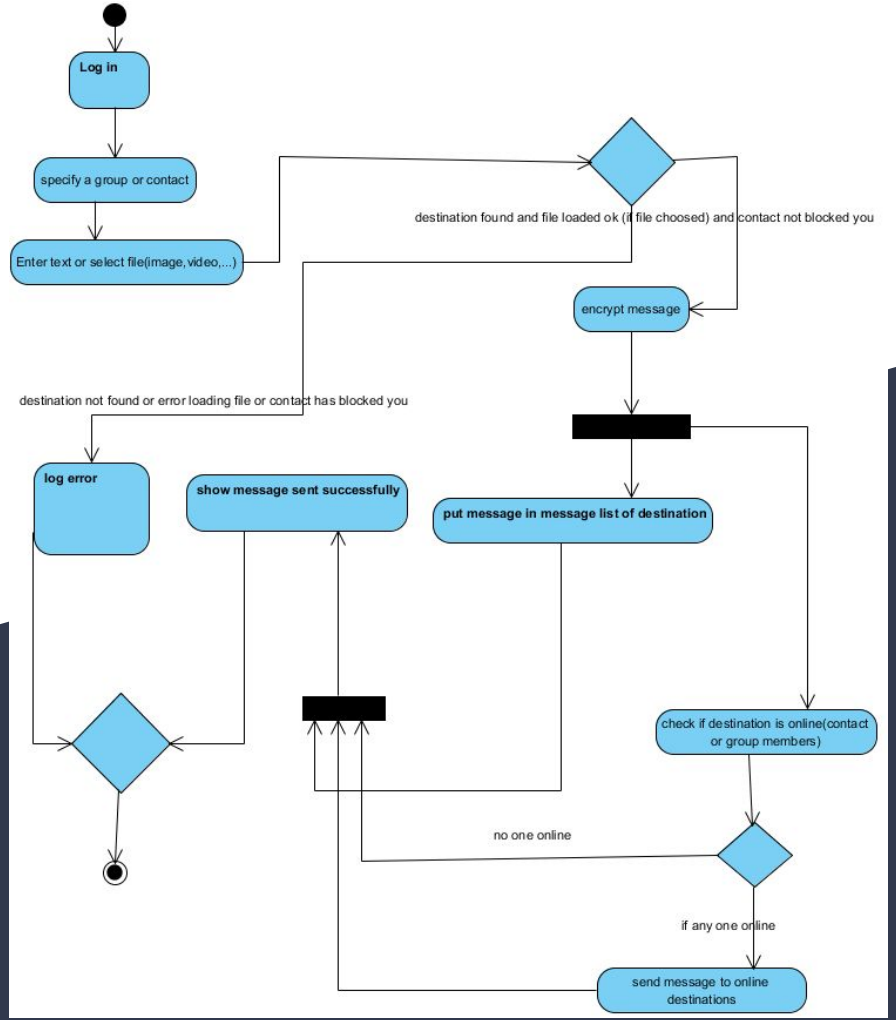
Login Activity Diagram



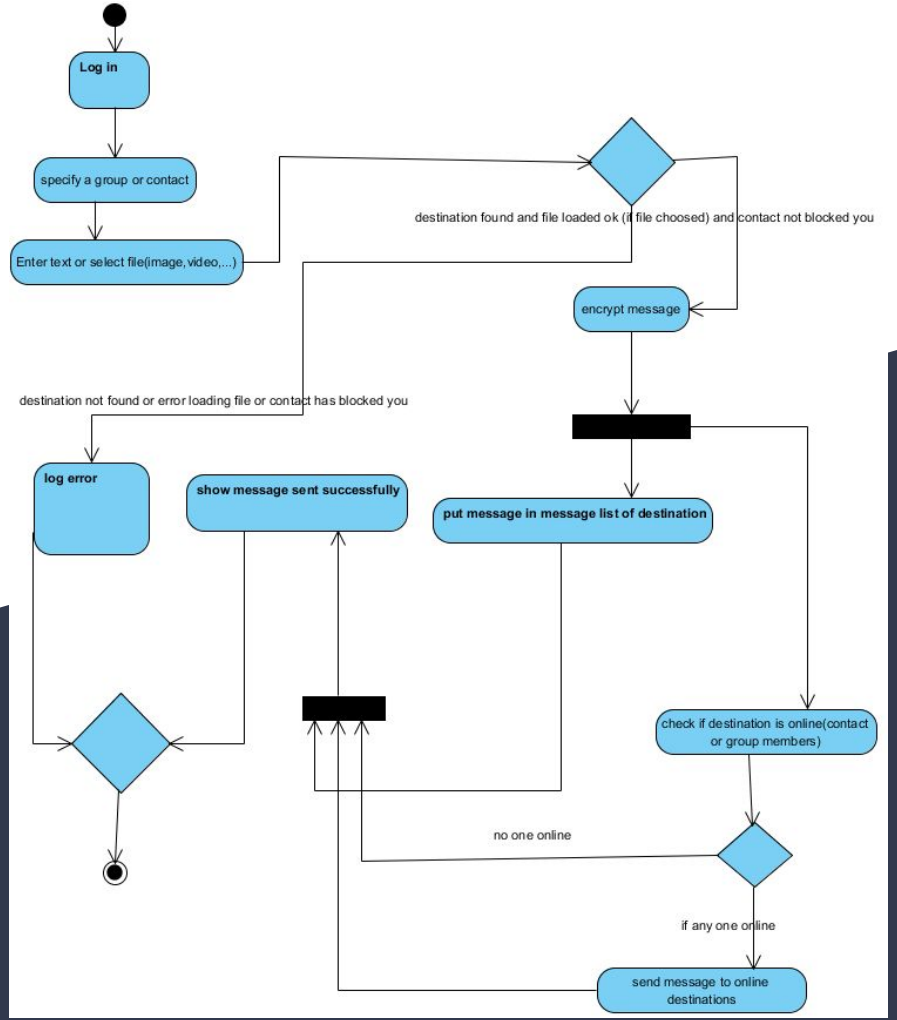
Logout Activity Diagram

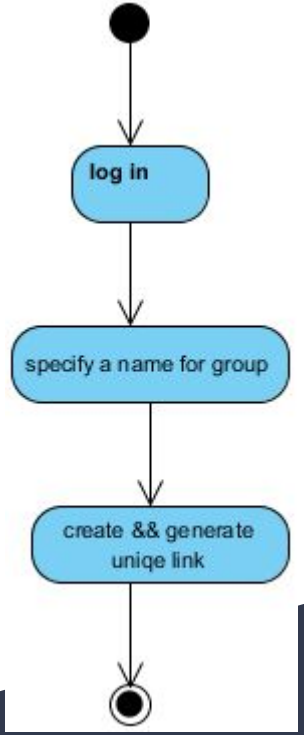


Send Message to User Activity Diagram



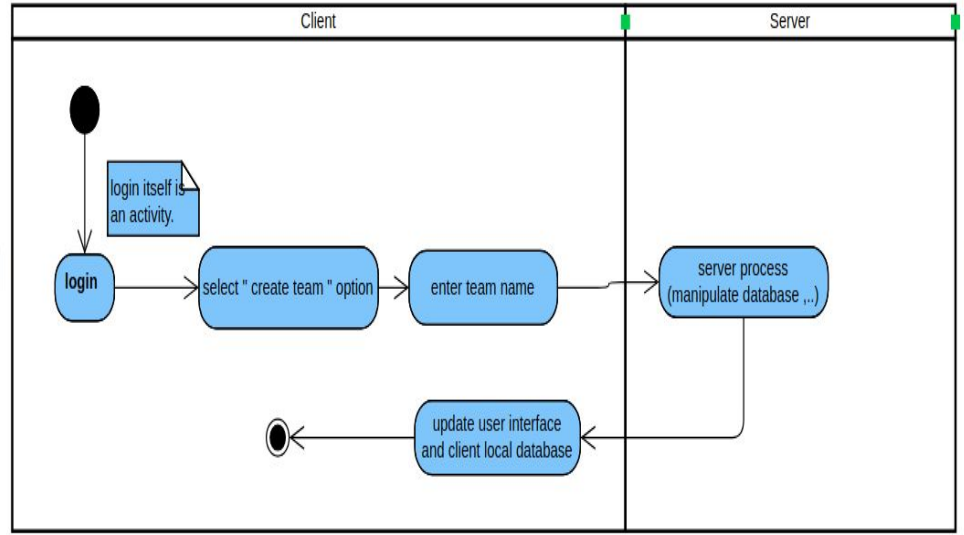
Send Message to Group Activity Diagram



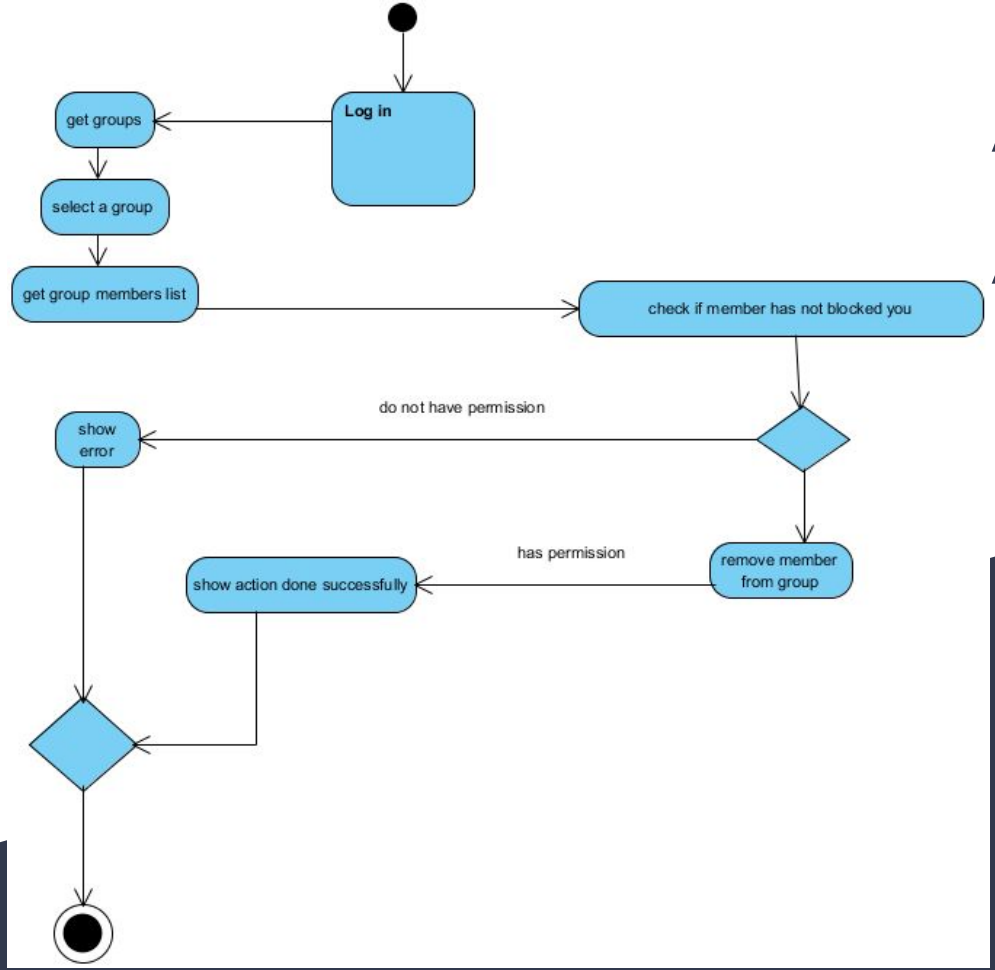


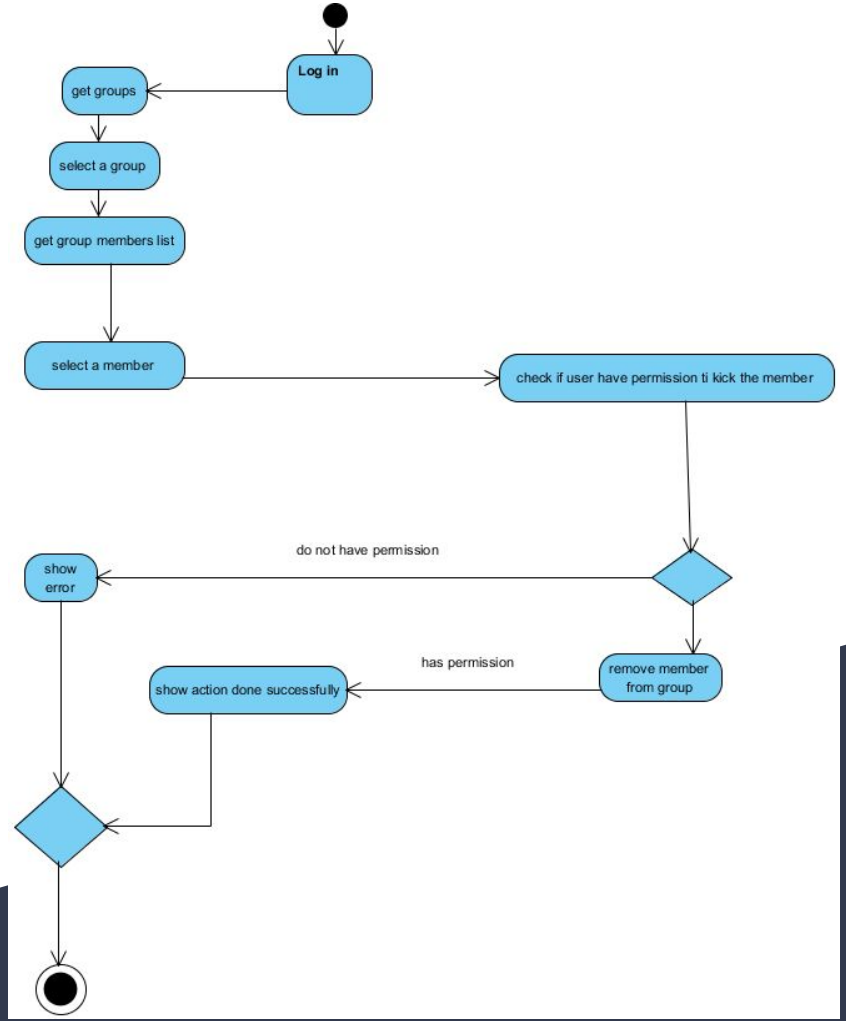
Create Group Activity Diagram

Create Team Activity Diagram



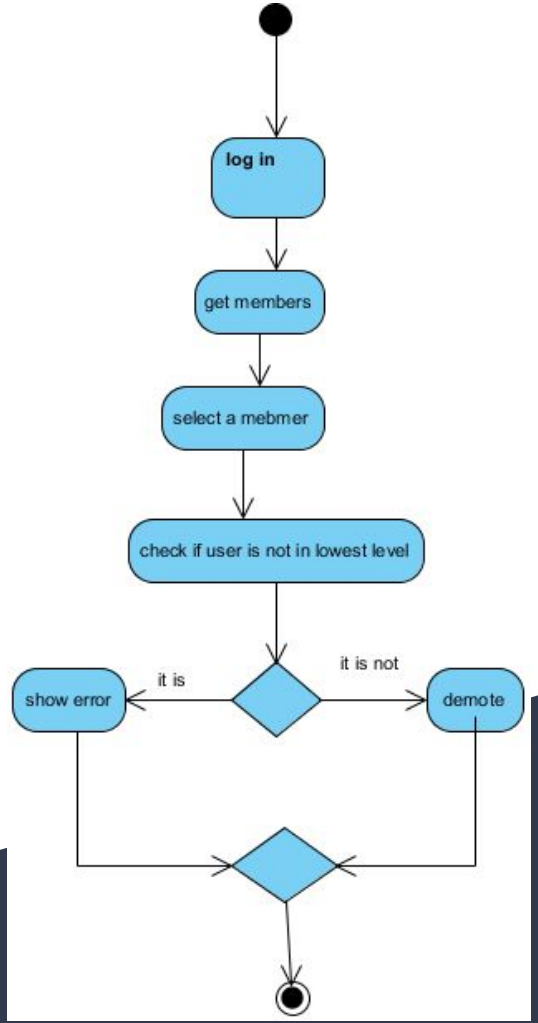
Add Member Activity Diagram

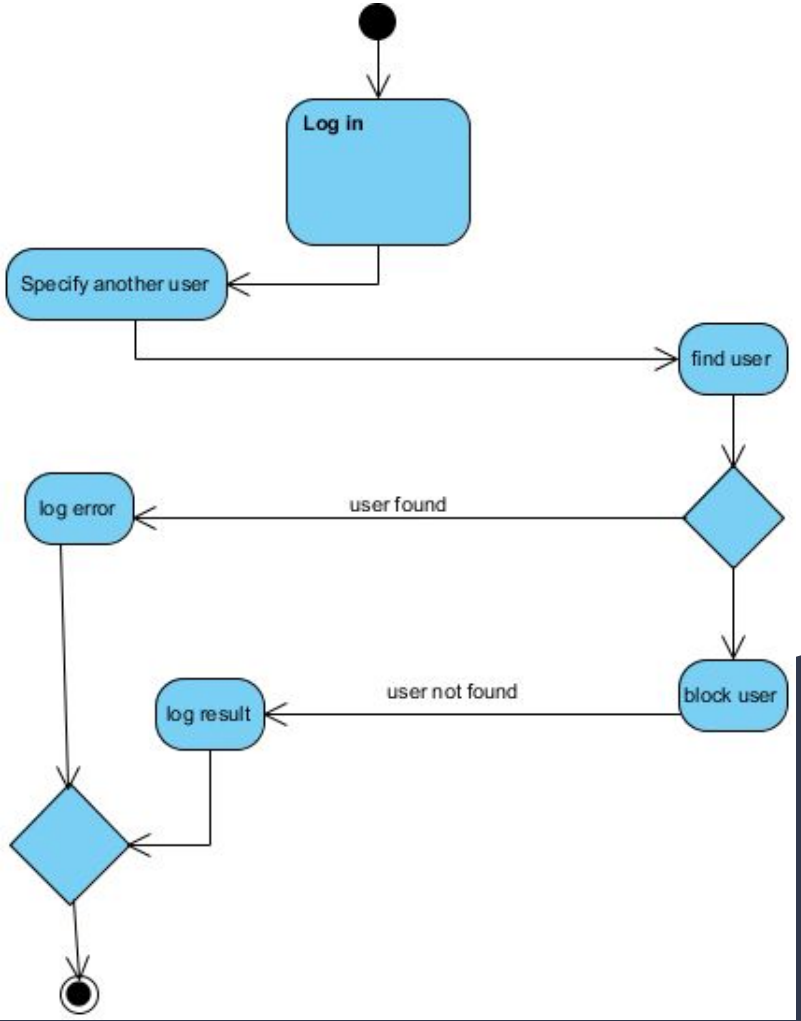




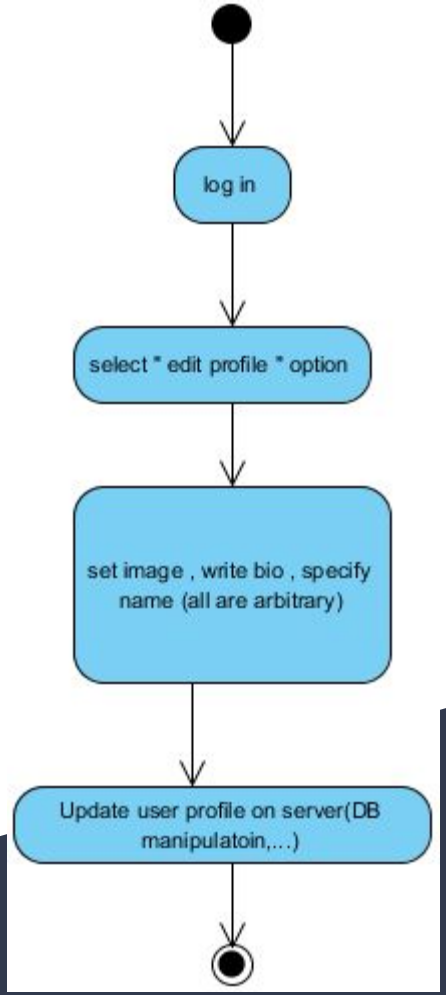
Kick Out Activity Diagram

Add Permission Activity Diagram



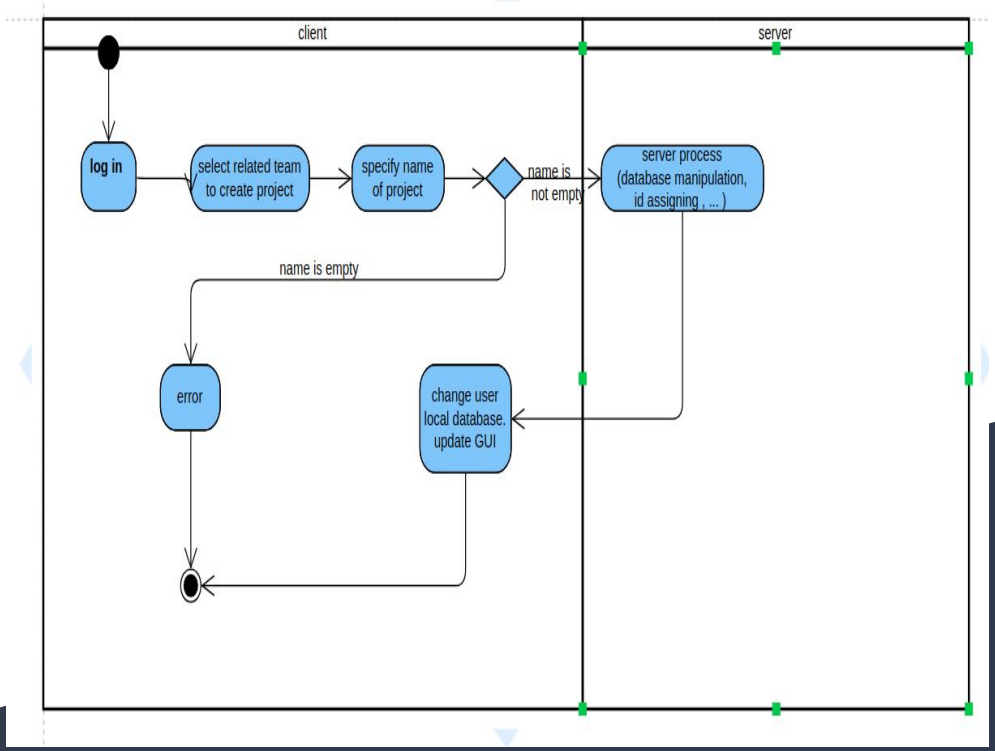


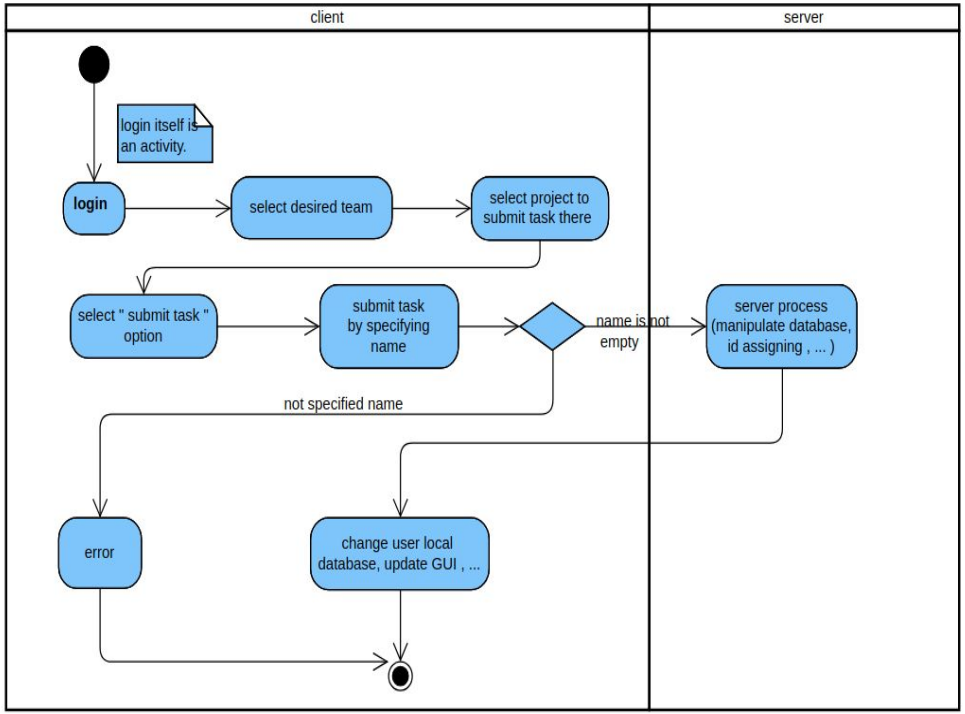
Block User Activity Diagram



Edit Profile Activity Diagram

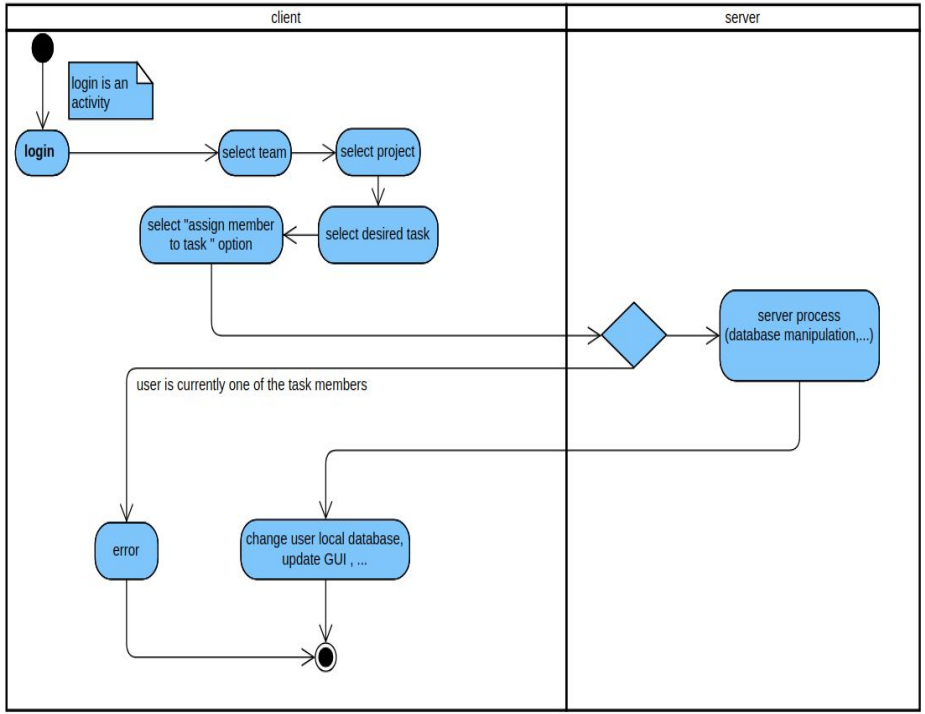
Create Project Activity Diagram



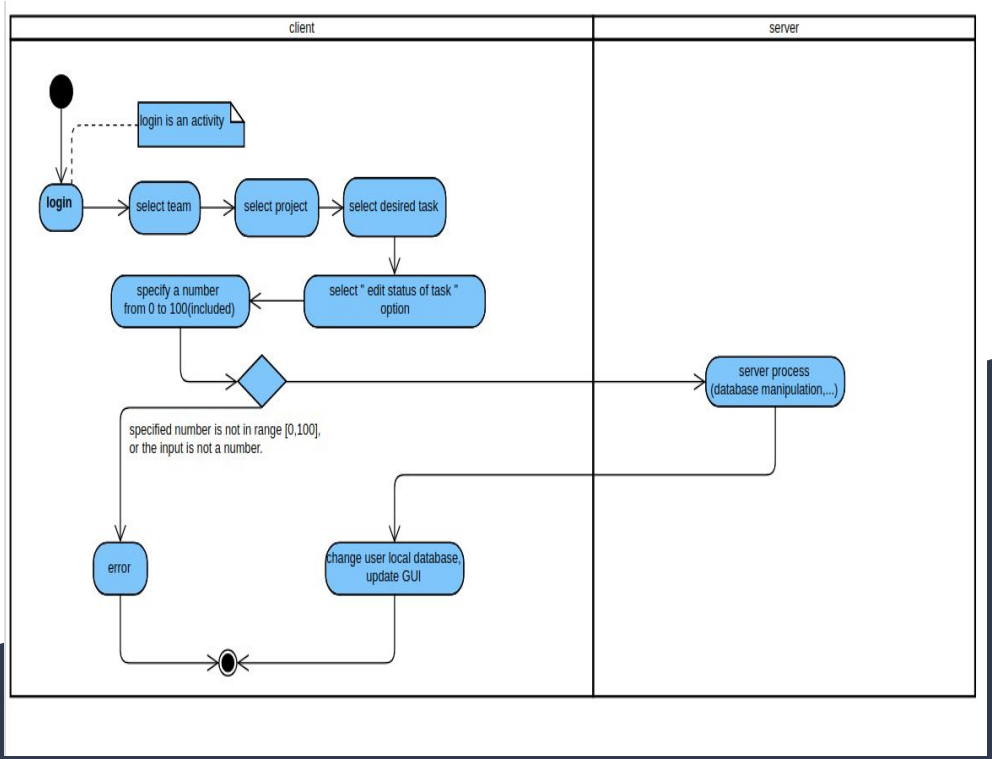


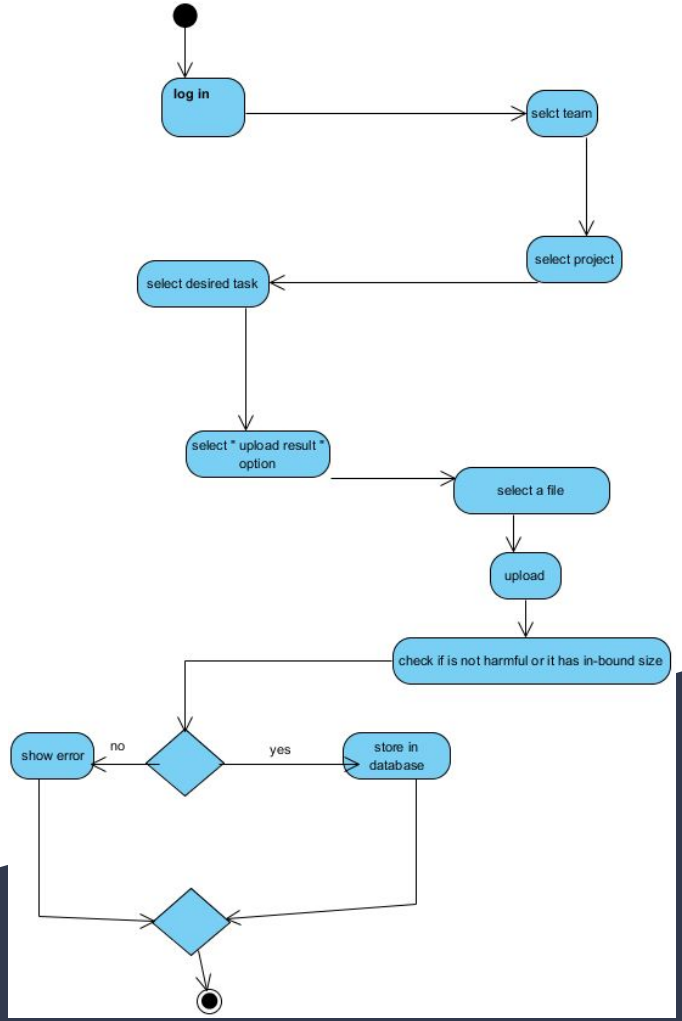
Submit Task Activity Diagram

Assign Member to Task Activity Diagram



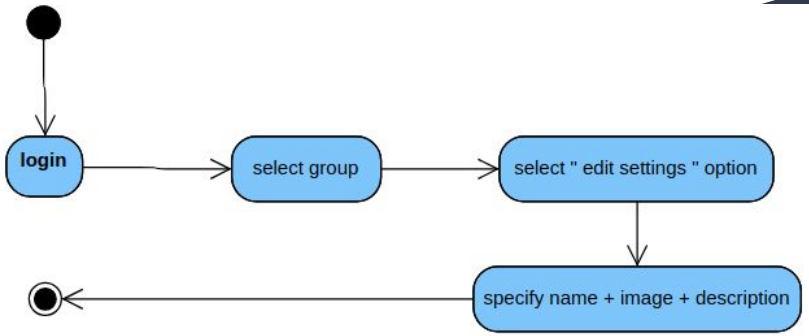
Edit Status of Task Activity Diagram





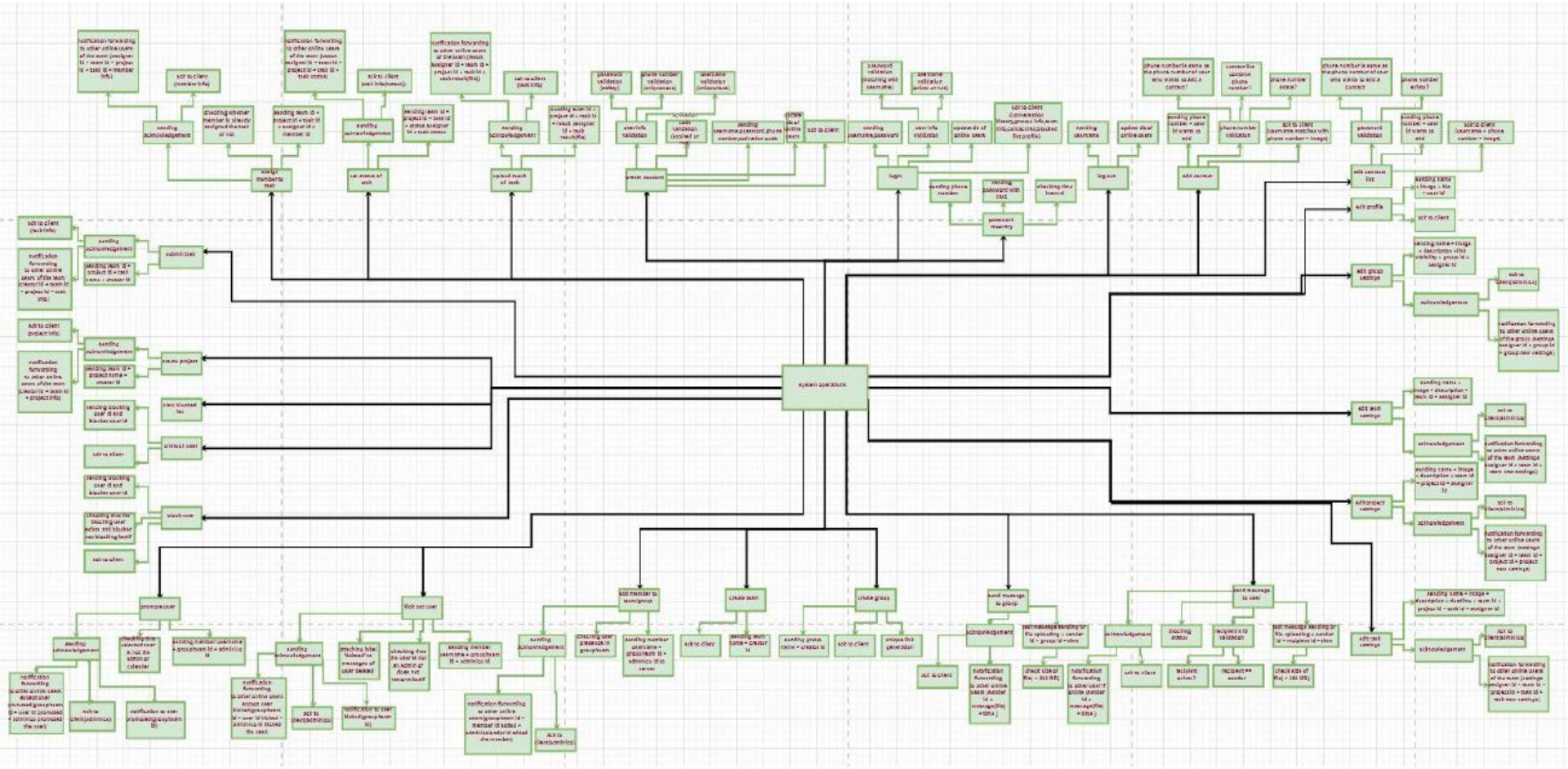
Upload Result of Task Activity Diagram

Edit Group Setting Activity Diagram

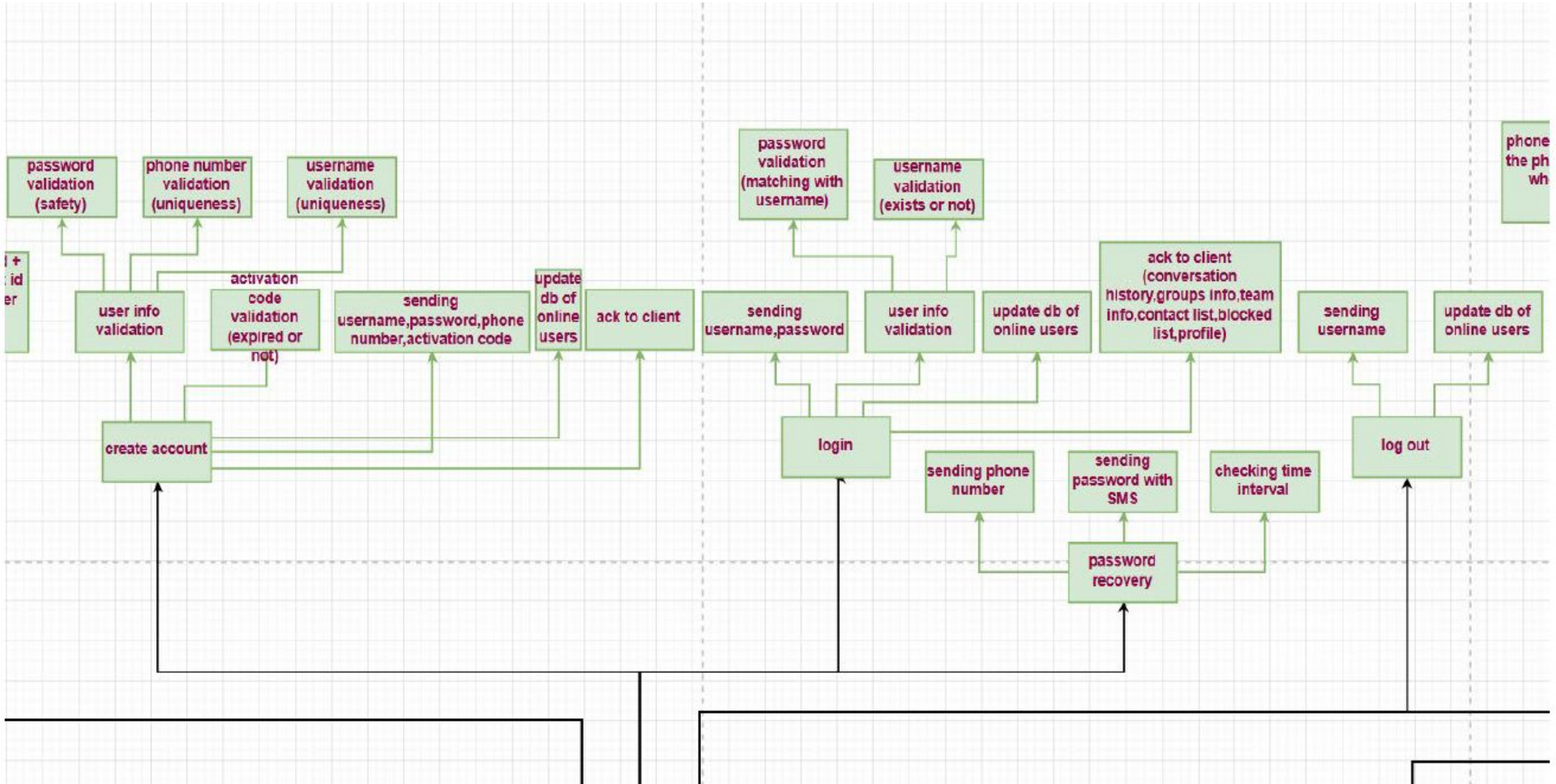


Team,Project,Task(DeadLine)

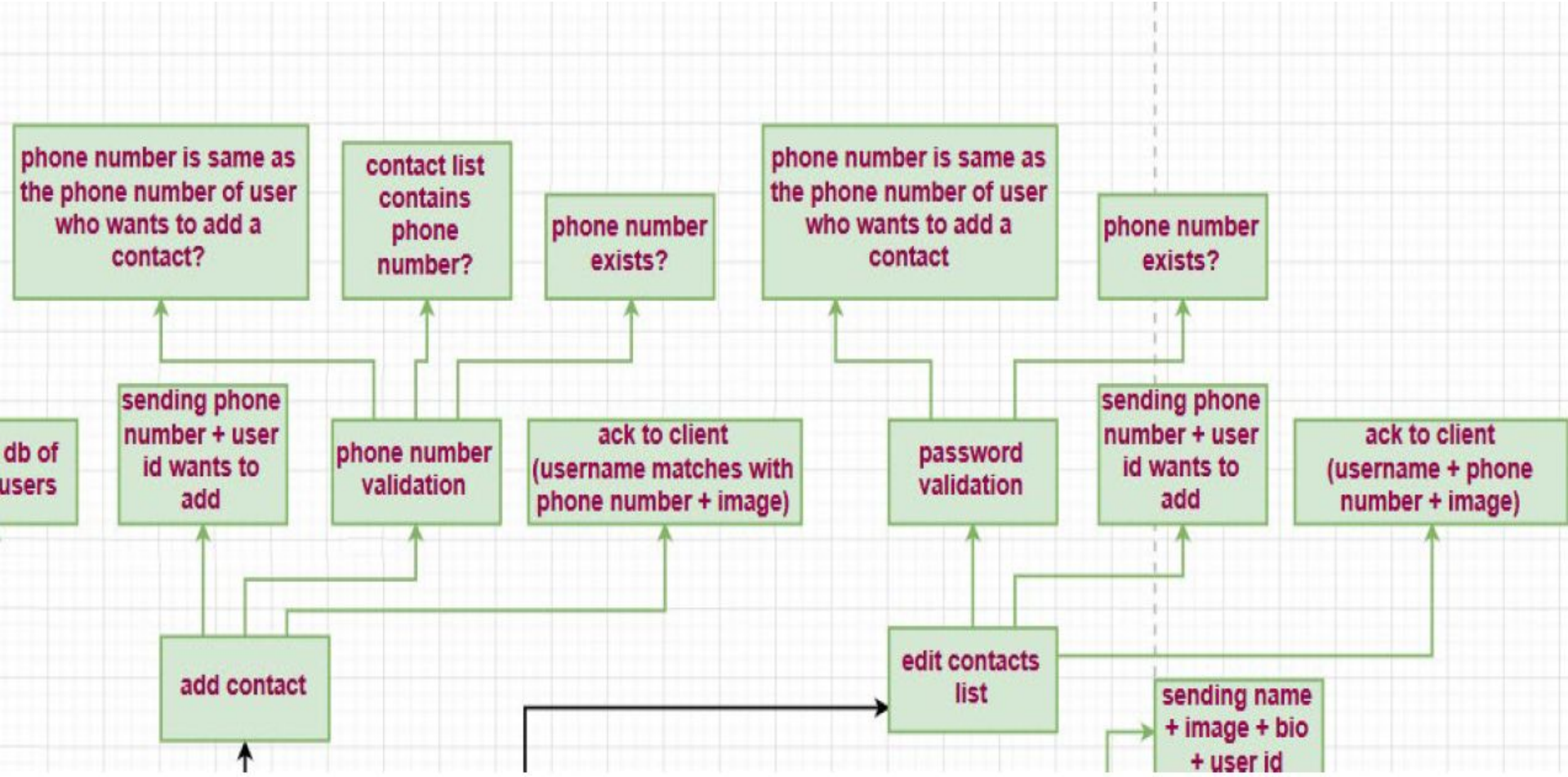
Functional Decomposition Diagram



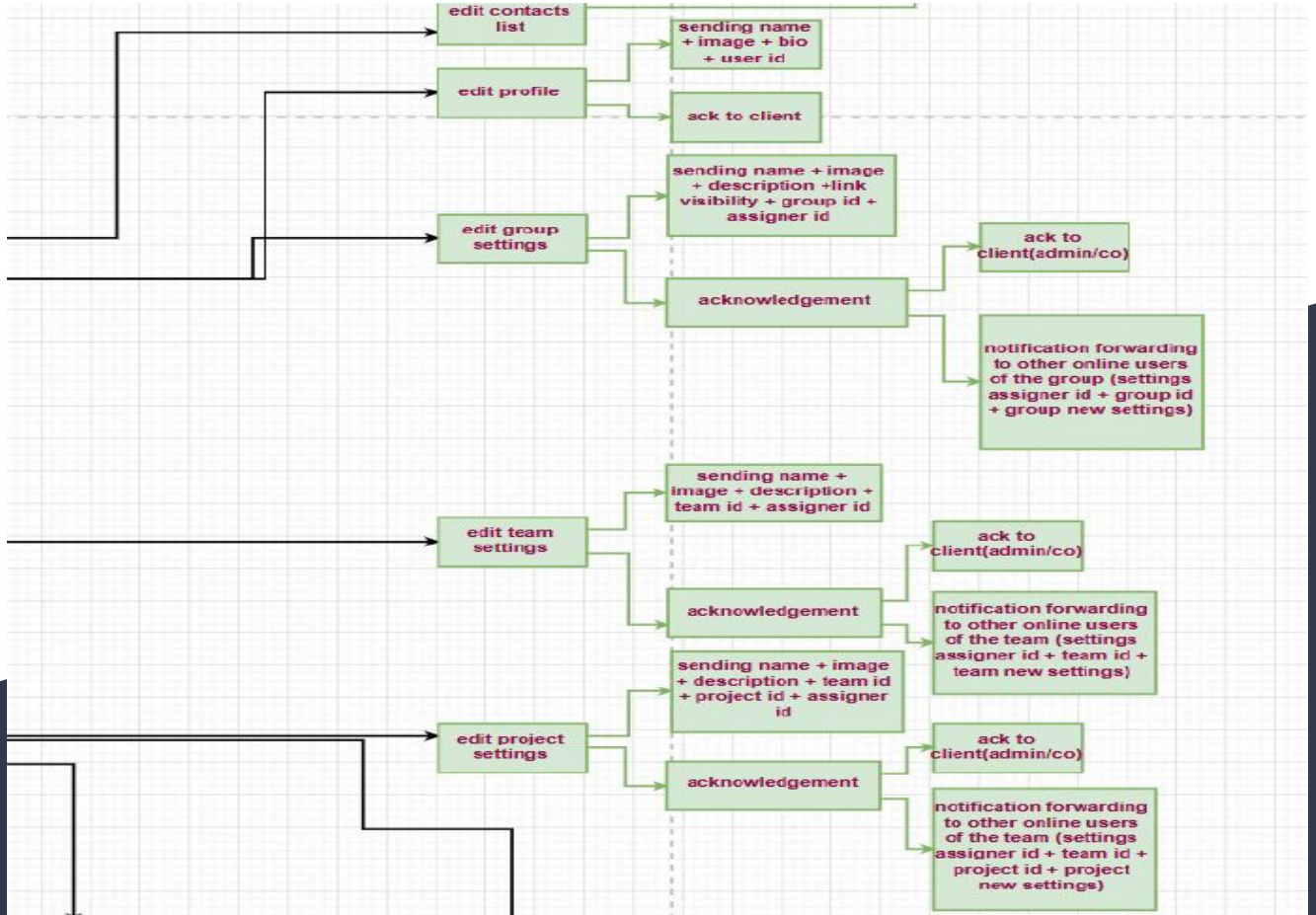
Details of FDD



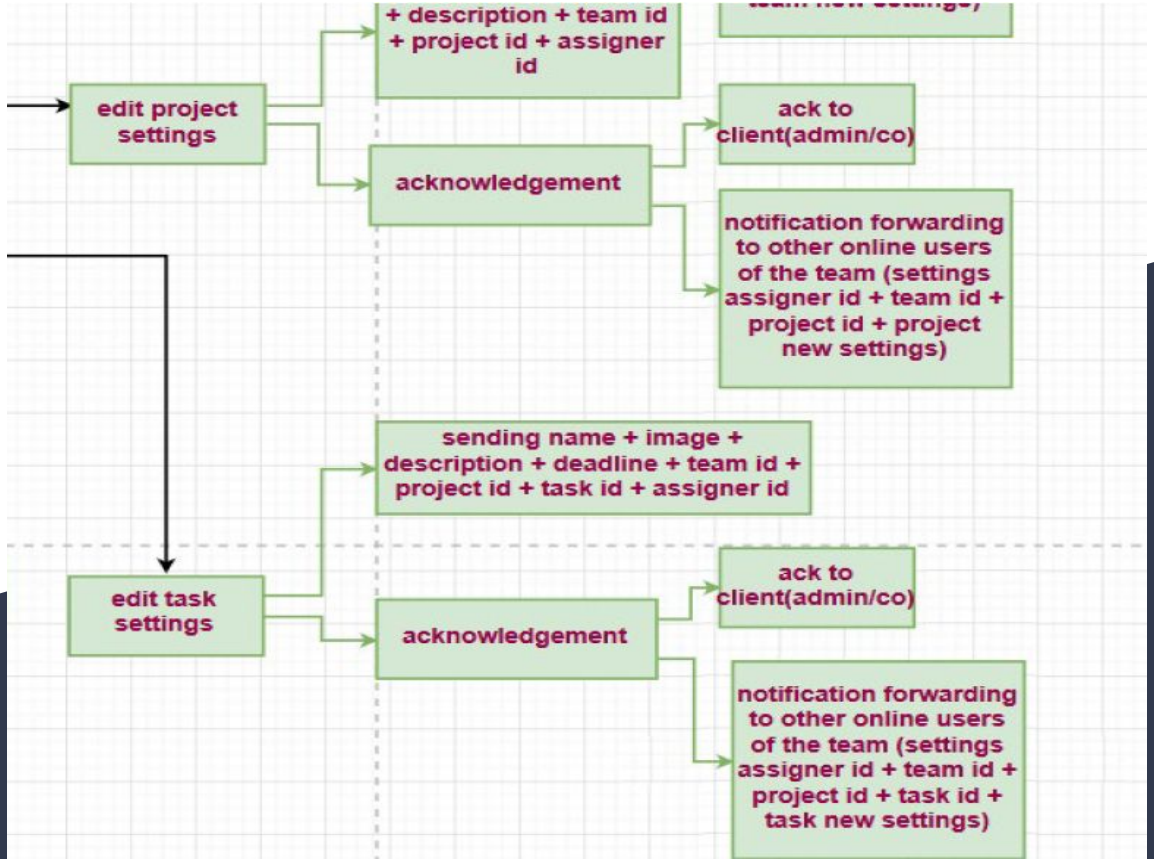
continue


phone number is same as the phone number of user who wants to add a contact?
contact list contains phone number?
phone number exists?
phone number is same as the phone number of user who wants to add a contact
phone number exists?
db of users
sending phone number + user id wants to add
phone number validation
ack to client (username matches with phone number + image)
password validation
sending phone number + user id wants to add
ack to client (username + phone number + image)
add contact
edit contacts list
sending name + image + bio + user id

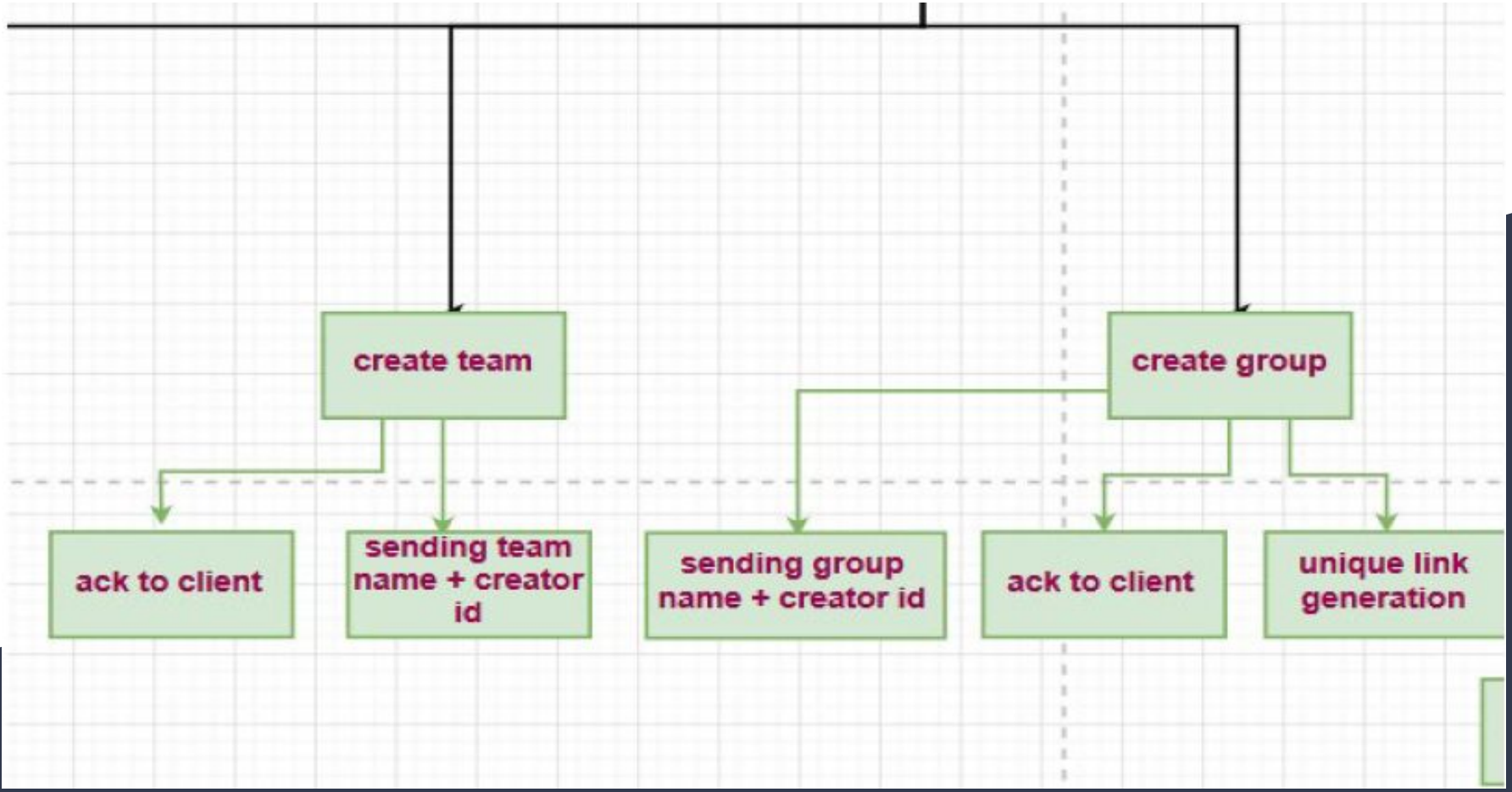
continue


edit contacts list
sending name + image + bio + user id
edit profile
ack to client
sending name + image + description +link visibility + group id + assigner id
edit group settings
ack to client(admin/co)
acknowledgement
notification forwarding to other online users of the group (settings assigner id + group id + group new settings)
sending name + image + description + team id + assigner id
edit team settings
ack to client(admin/co)
acknowledgement
notification forwarding to other online users of the team (settings assigner id + team id + team new settings)
sending name + image + description + team id + project id + assigner id
edit project settings
ack to client(admin/co)
acknowledgement
notification forwarding to other online users of the team (settings assigner id + team id + project id + project new settings)

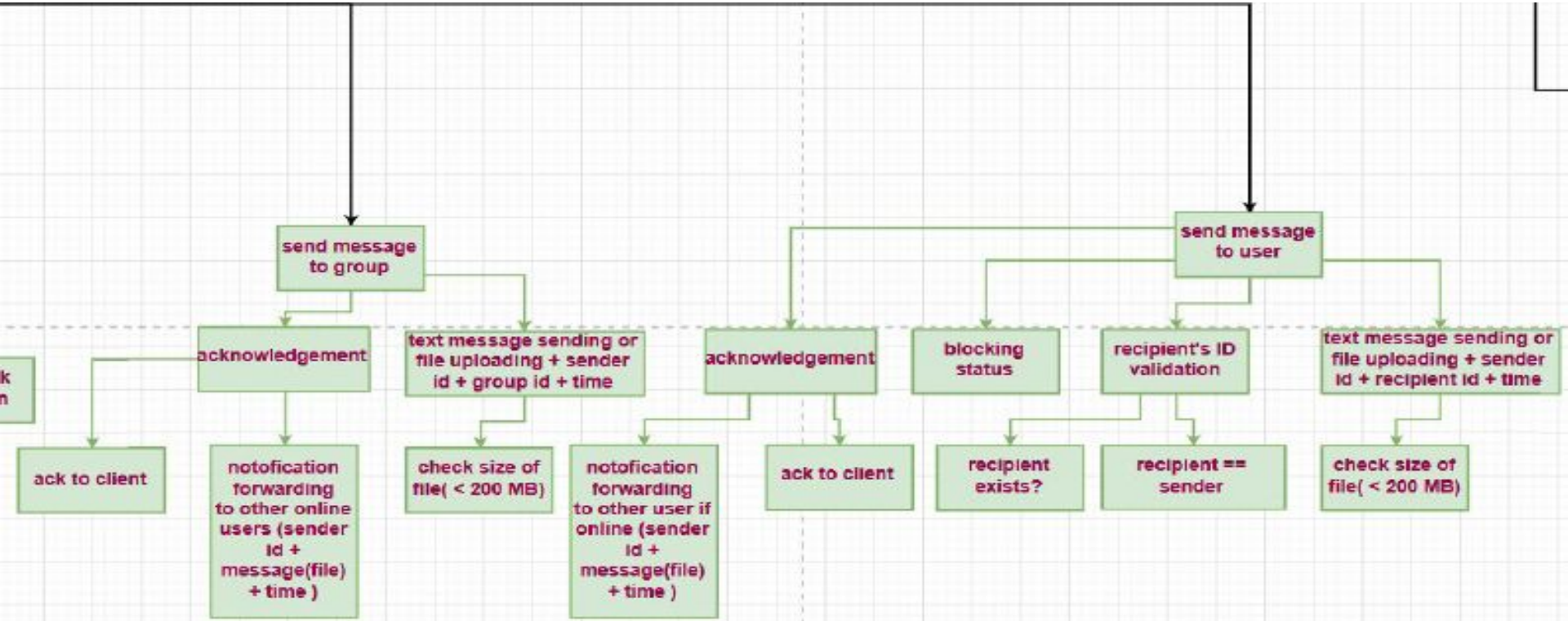
continue

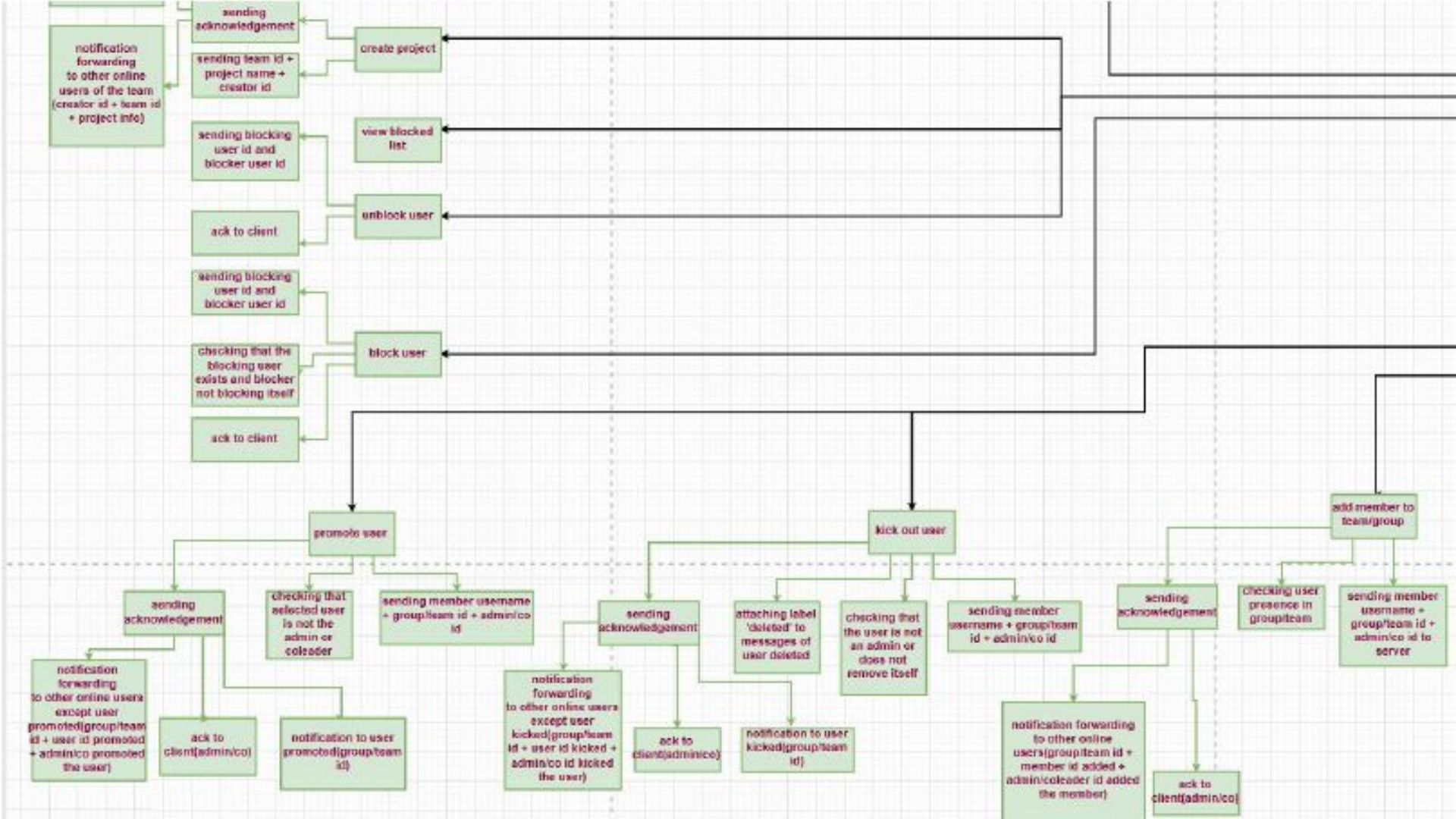

+ description + team id + project id + assigner id
edit project settings
ack to client(admin/co)
acknowledgement
notification forwarding to other online users of the team (settings assigner id + team id + project id + project new settings)
sending name + image + description + deadline + team id + project id + task id + assigner id
edit task settings
ack to client(admin/co)
acknowledgement
notification forwarding to other online users of the team (settings assigner id + team id + project id + task id + task new settings)

continue

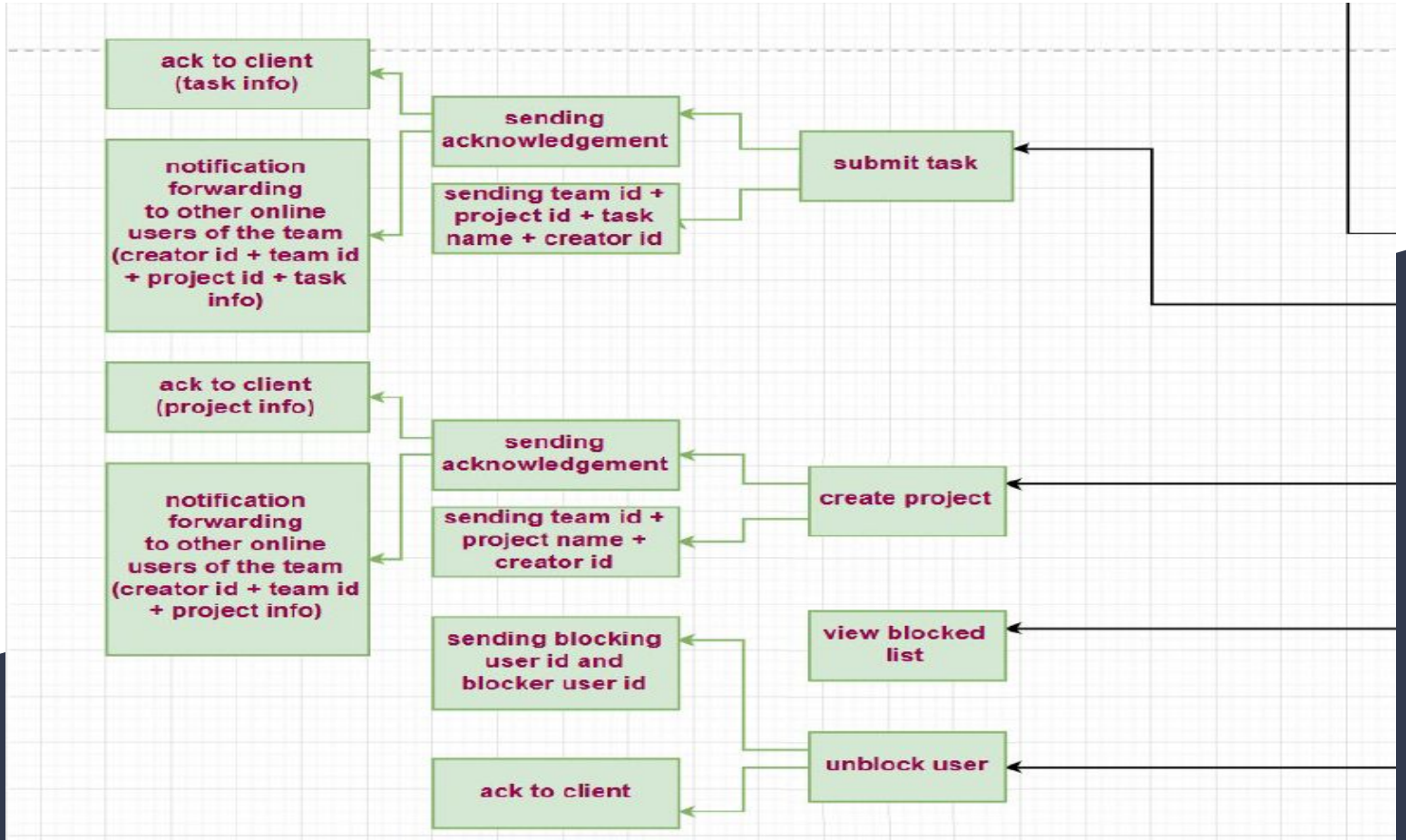

create team
create group
ack to client
sending team name + creator id
sending group name + creator id
ack to client
unique link generation

continue

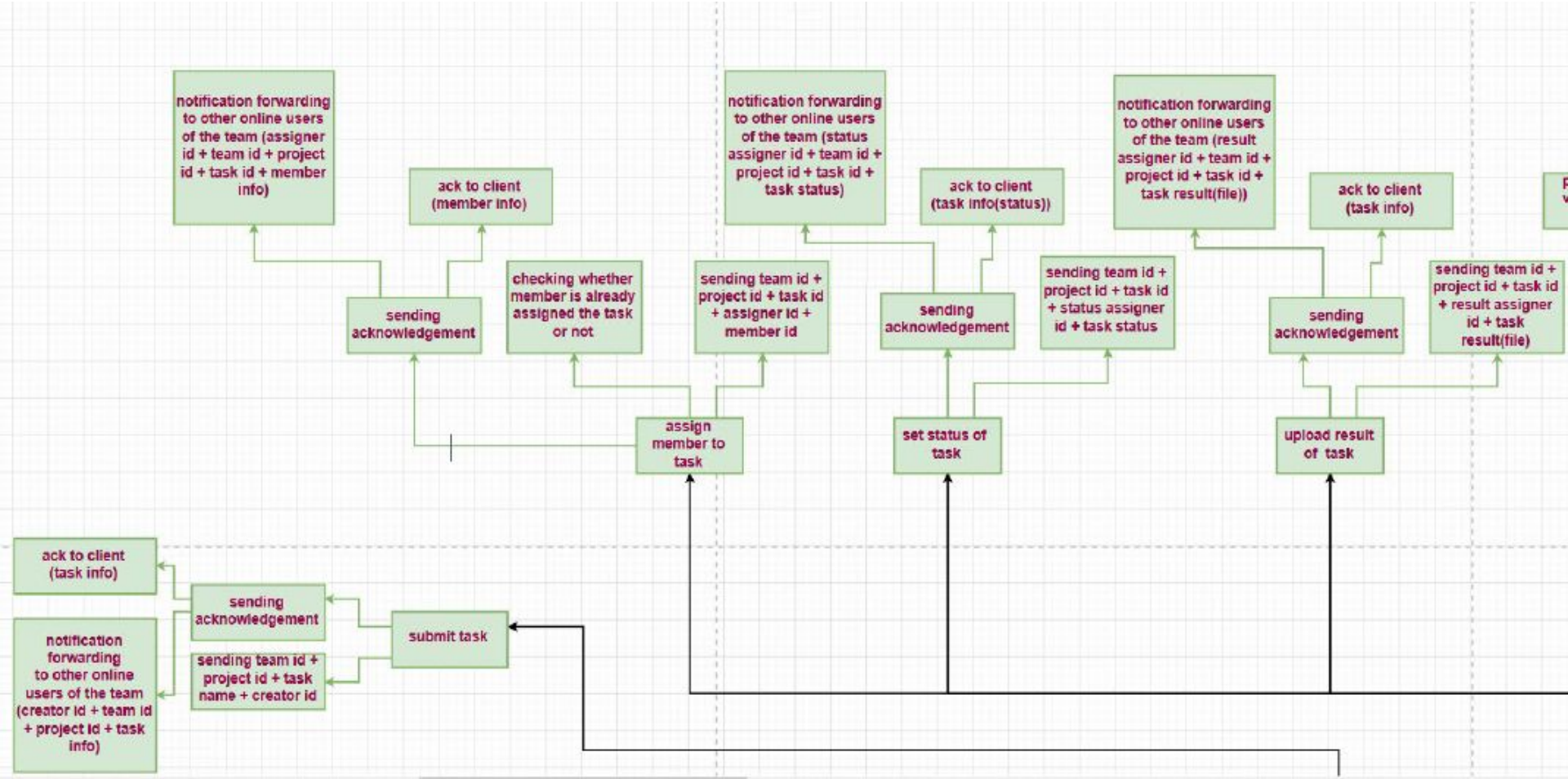

send message to group
send message to user
acknowledgement
text message sending or file uploading + sender id + group id + time
acknowledgement
blocking status
recipient's ID validation
text message sending or file uploading + sender id + recipient id + time
ack to client
notofication forwarding to other online users (sender id + message(file) + time)
check size of file(< 200 MB)
notofication forwarding to other user if online (sender id + message(file) + time)
ack to client
recipient exists?
recipient == sender
check size of file(< 200 MB)


sending acknowledgement
notification forwarding to other online users of the team (creator id + team id + project info)
sending team id + project name + creator id
create project
sending blocking user id and blocker user id
view blocked list
ack to client
unblock user
sending blocking user id and blocker user id
checking that the blocking user exists and blocker not blocking itself
ack to client
block user
promote user
kick out user
add member to team/group
sending acknowledgement
checking that selected user is not the admin or coleader
sending member username + group/team id + admin/co id
notification forwarding to other online users except user promoted(group/team id + user id promoted + admin/co promoted the user)
ack to client(admin/co)
notification to user promoted(group/team id)
sending acknowledgement
attaching label 'deleted' to messages of user deleted
checking that the user is not an admin or does not remove itself
sending member username + group/team id + admin/co id
notification forwarding to other online users except user kicked(group/team id + user id kicked + admin/co id kicked the user)
ack to client(admin/co)
notification to user kicked(group/team id)
sending acknowledgement
checking user presence in group/team
sending member username + group/team id + admin/co id to server
notification forwarding to other online users(group/team id + member id added + admin/coleader id added the member)
ack to client(admin/co)

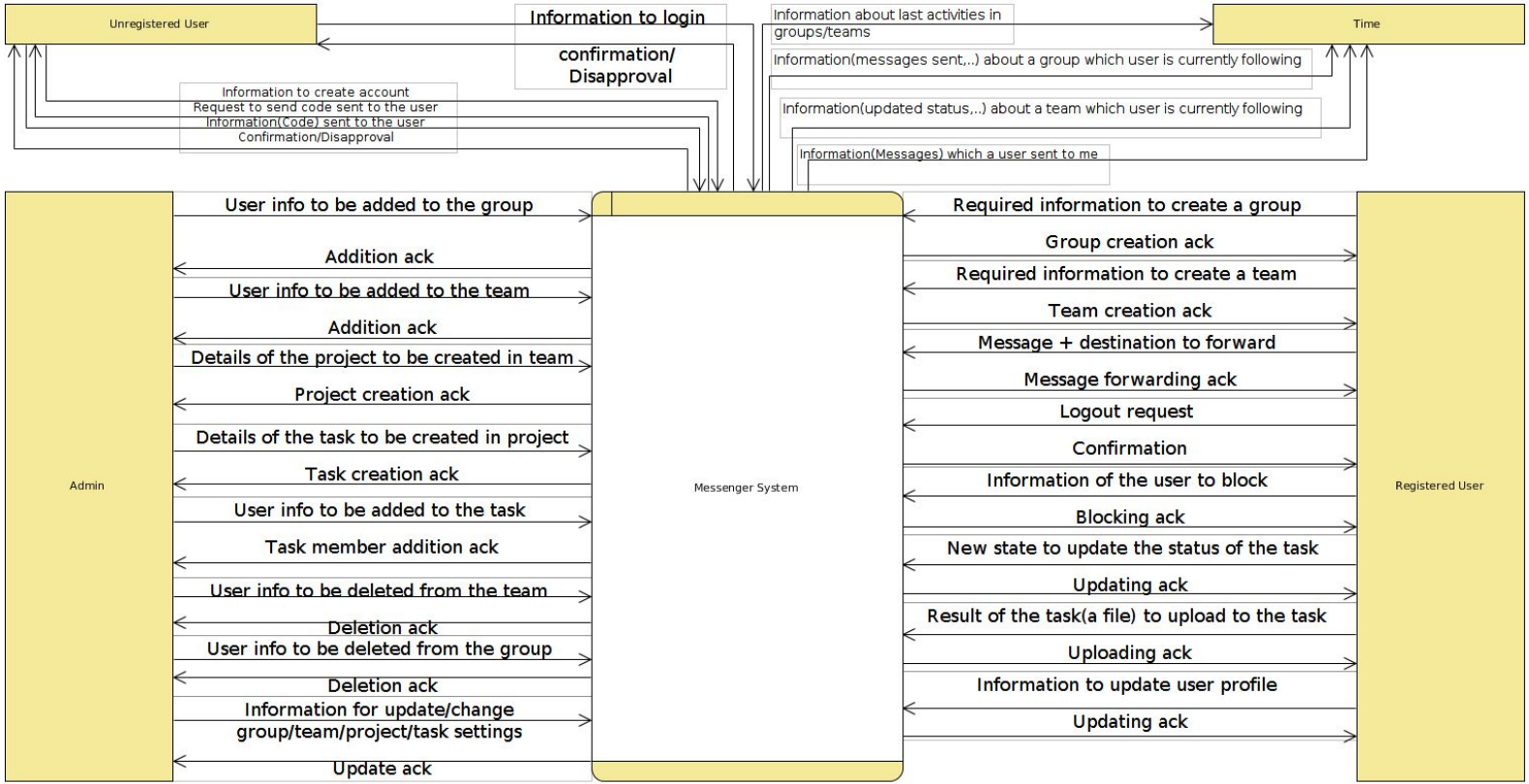
continue


ack to client (task info)
sending acknowledgement
notification forwarding to other online users of the team (creator id + team id + project id + task info)
sending team id + project id + task name + creator id
submit task
ack to client (project info)
sending acknowledgement
notification forwarding to other online users of the team (creator id + team id + project info)
sending team id + project name + creator id
create project
sending blocking user id and blocker user id
view blocked list
ack to client
unblock user

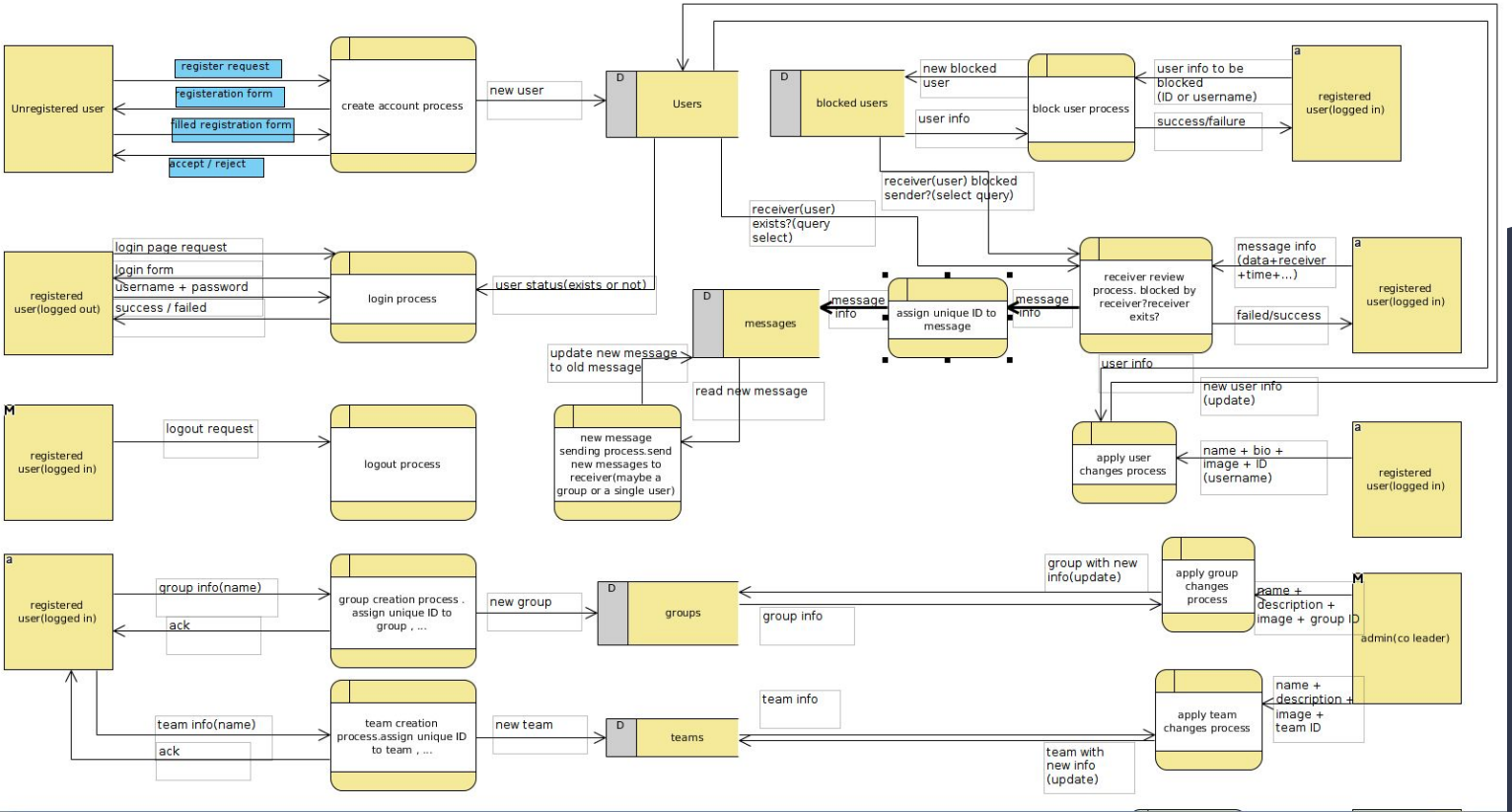
continue


notification forwarding to other online users of the team (assigner id + team id + project id + task id + member info)
ack to client (member info)
notification forwarding to other online users of the team (status assigner id + team id + project id + task id + task status)
ack to client (task info(status))
notification forwarding to other online users of the team (result assigner id + team id + project id + task id + task result(file))
ack to client (task info)
sending acknowledgement
checking whether member is already assigned the task or not
sending team id + project id + task id + assigner id + member id
sending acknowledgement
sending team id + project id + task id + status assigner id + task status
sending acknowledgement
sending team id + project id + task id + result assigner id + task result(file)
assign member to task
set status of task
upload result of task
ack to client (task info)
sending acknowledgement
notification forwarding to other online users of the team (creator id + team id + project id + task info)
submit task
sending team id + project id + task name + creator id

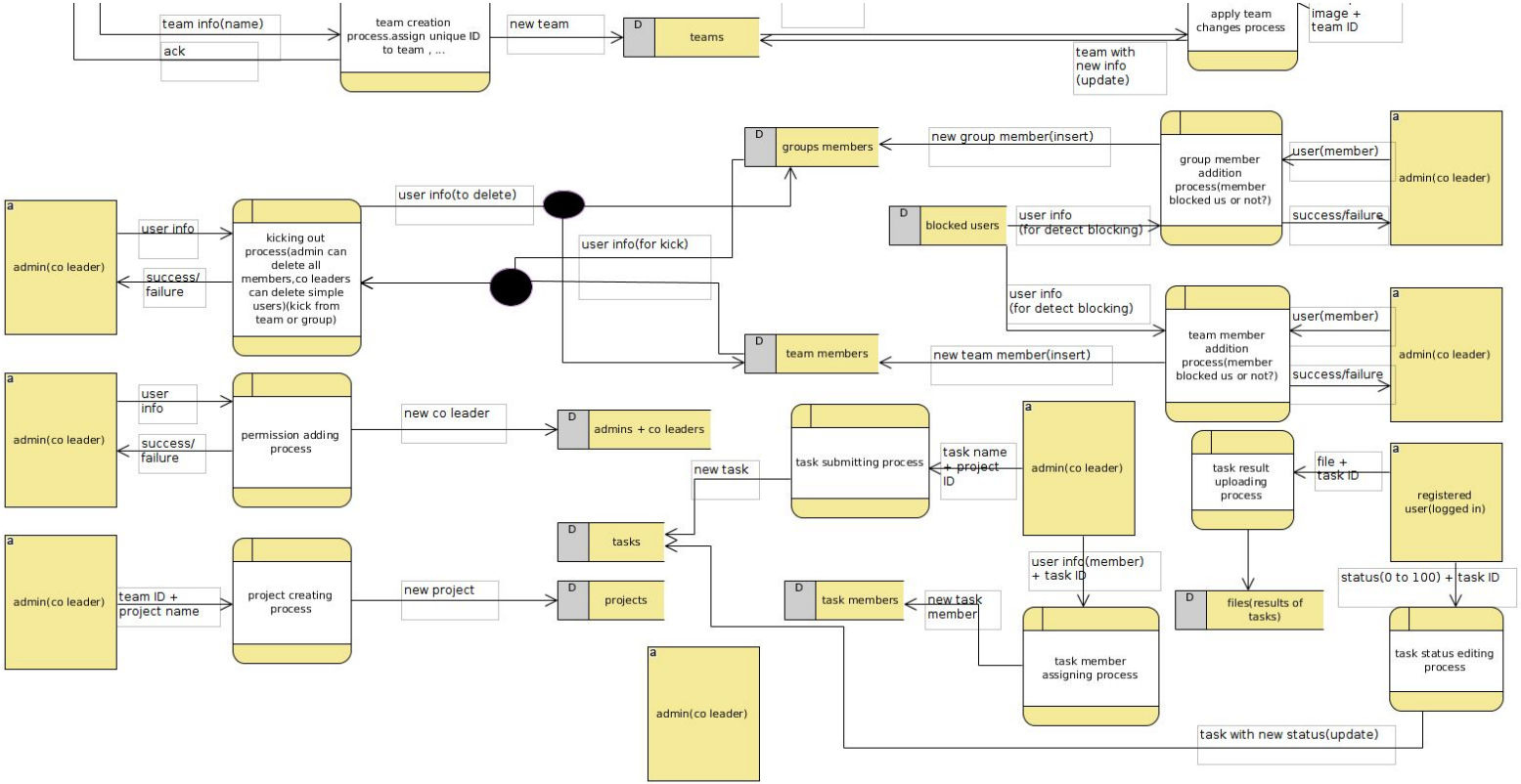
DFD Level 0


Unregistered User
Information to login
confirmation/
Disapproval
Information about last activities in groups/teams
Time
Information(messages sent,..) about a group which user is currently following
Information(updated status,..) about a team which user is currently following
Information(Messages) which a user sent to me
Information to create account
Request to send code sent to the user
Information(Code) sent to the user
Confirmation/Disapproval
Admin
User info to be added to the group
Addition ack
User info to be added to the team
Addition ack
Details of the project to be created in team
Project creation ack
Details of the task to be created in project
Task creation ack
User info to be added to the task
Task member addition ack
User info to be deleted from the team
Deletion ack
User info to be deleted from the group
Deletion ack
Information for update/change
group/team/project/task settings
Update ack
Messenger System
Required information to create a group
Group creation ack
Required information to create a team
Team creation ack
Message + destination to forward
Message forwarding ack
Logout request
Confirmation
Information of the user to block
Blocking ack
New state to update the status of the task
Updating ack
Result of the task(a file) to upload to the task
Uploading ack
Information to update user profile
Updating ack
Registered User

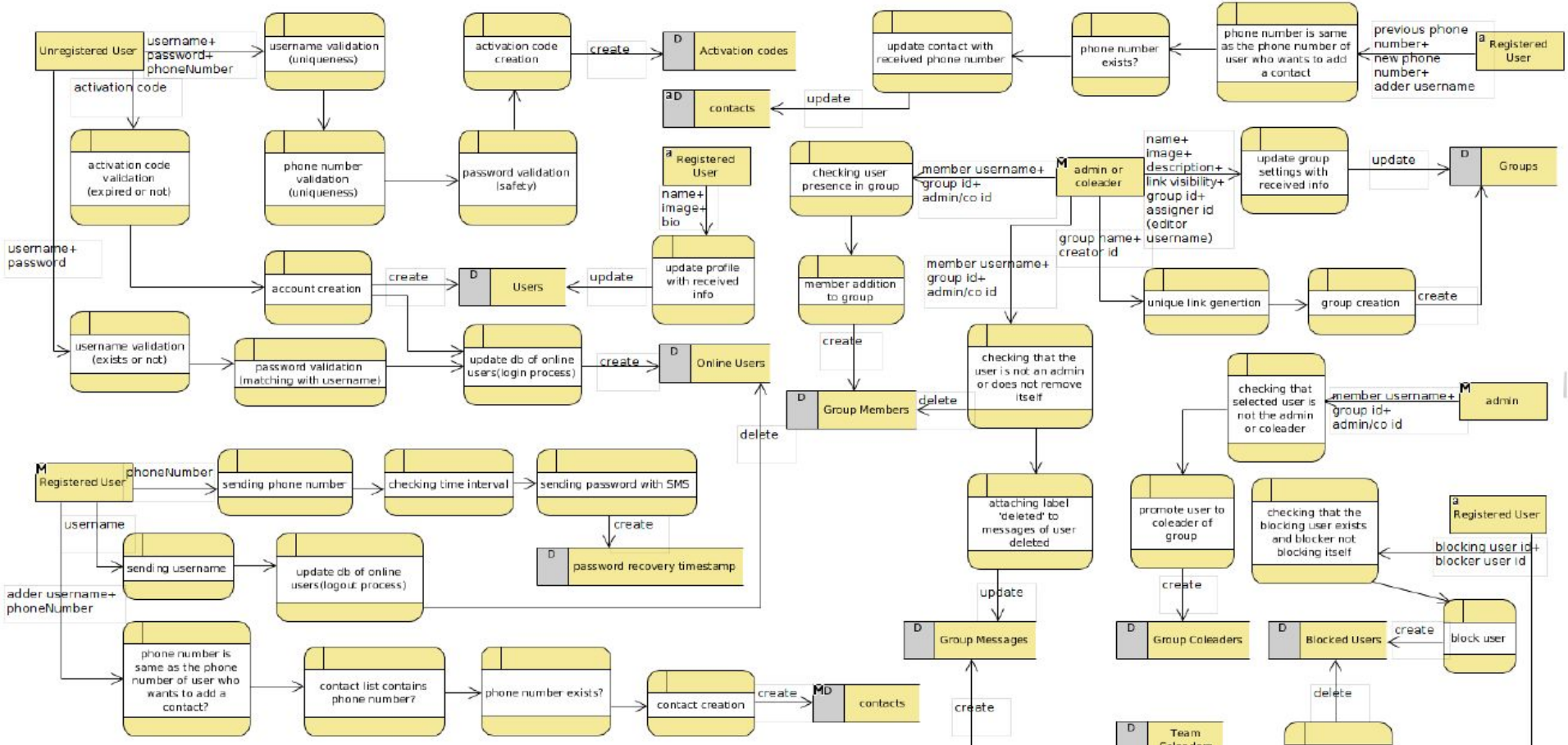
DFD Level 1 part 1



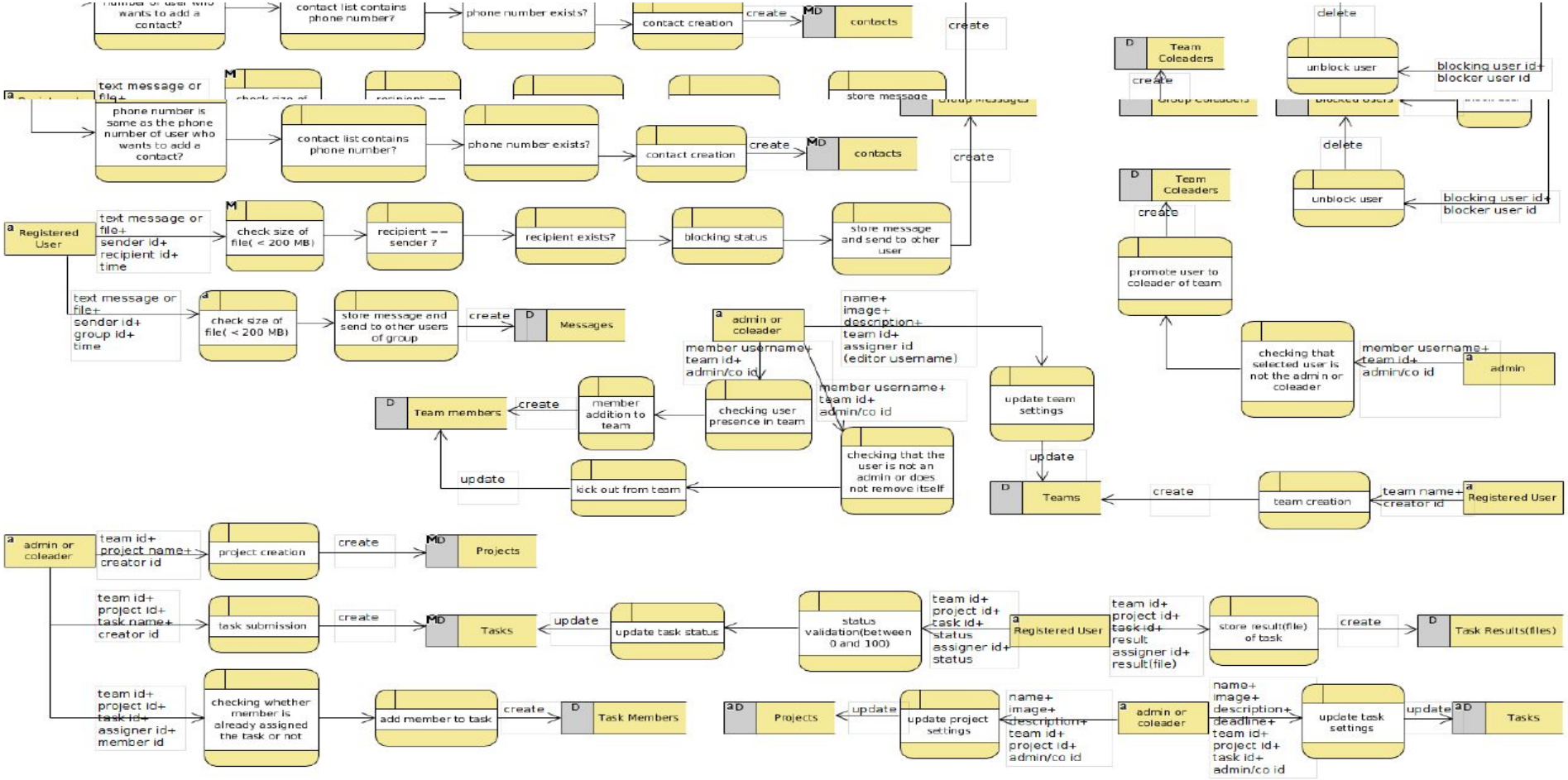
DFD Level 1 part 2



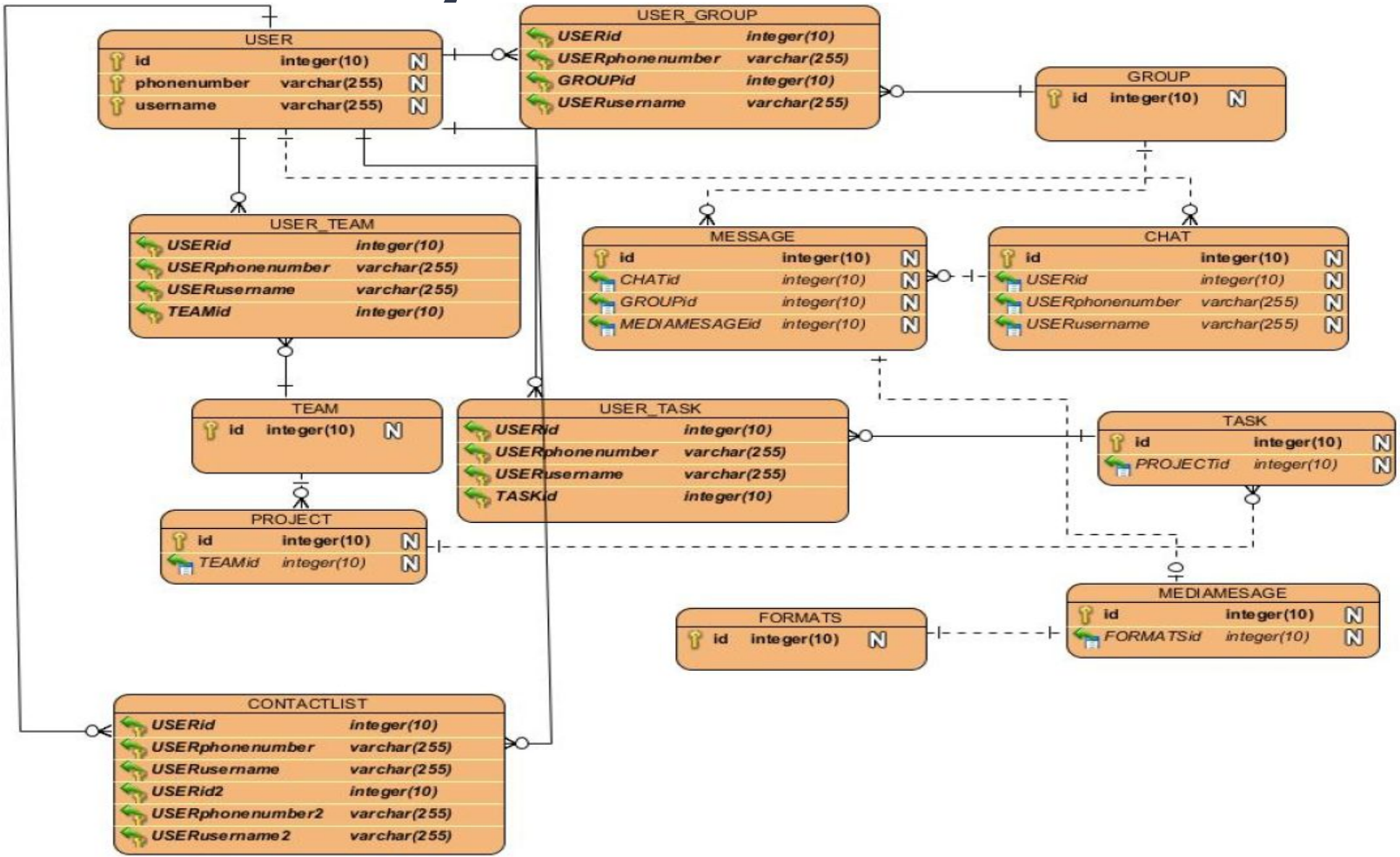
DFD level 2 part1



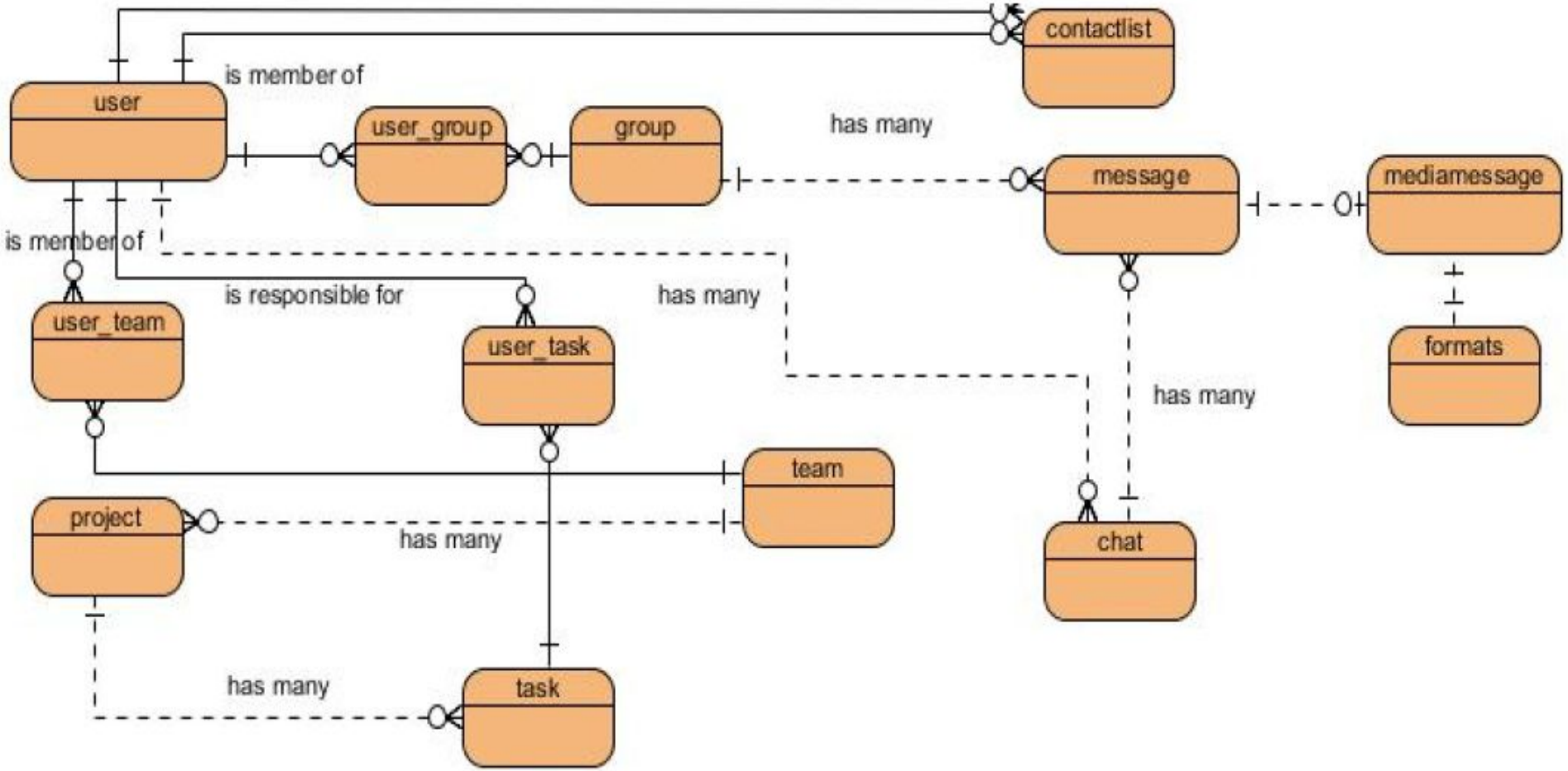
DFD level 2 part 2



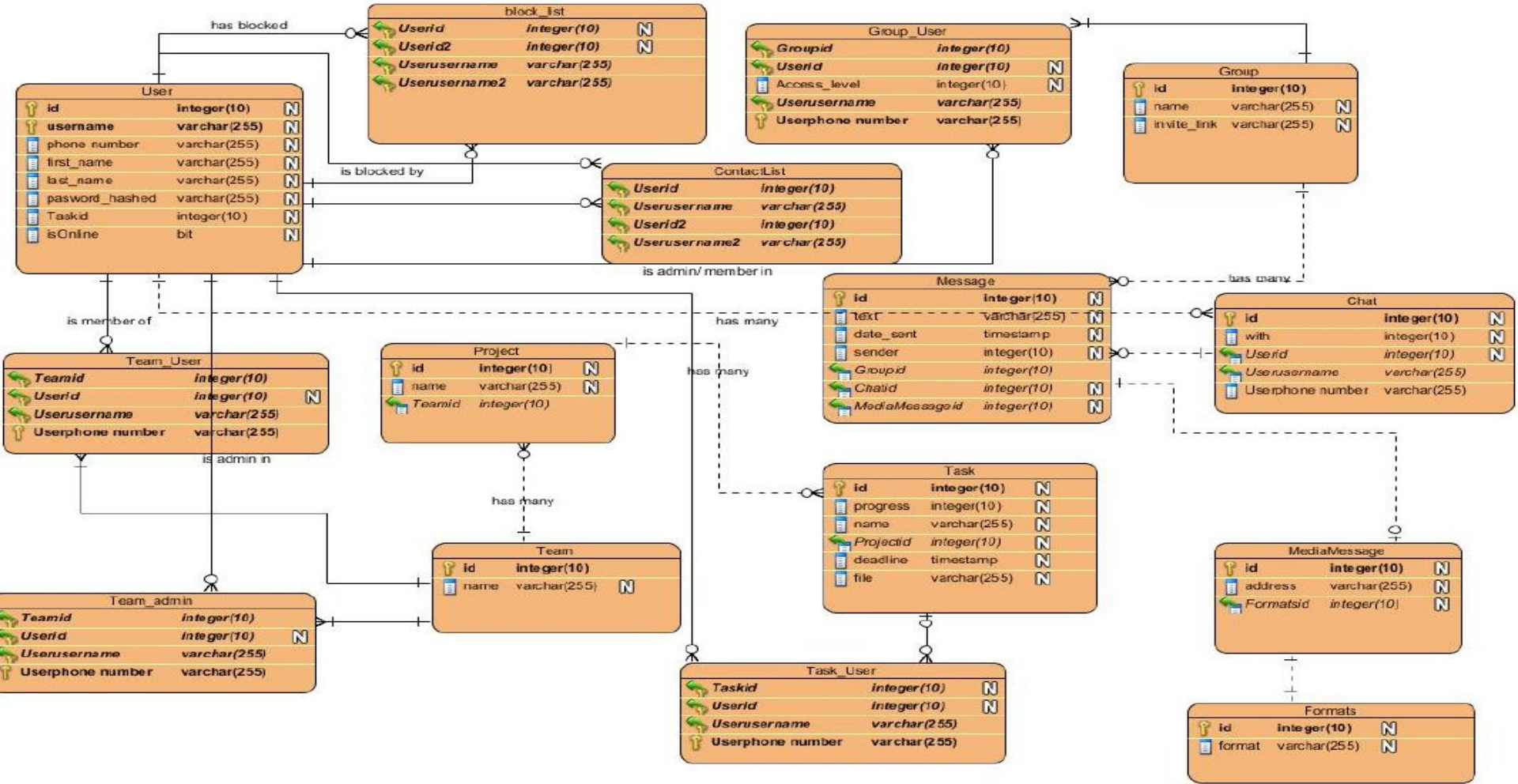
Key Based Data Model



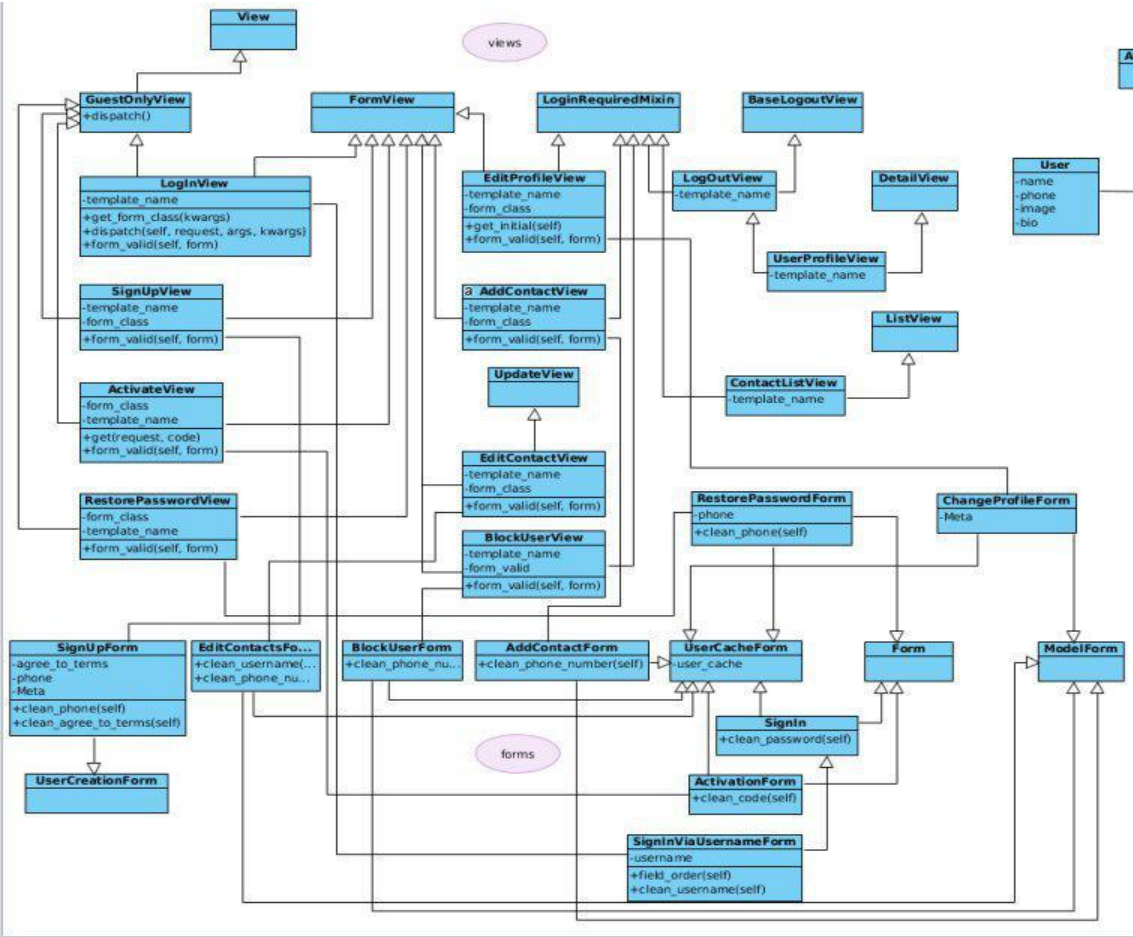
Context Based Data Model

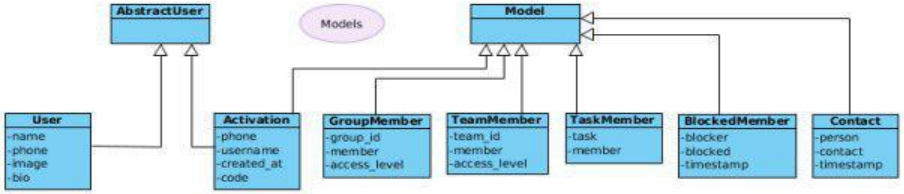


Fully Attributed Data Model

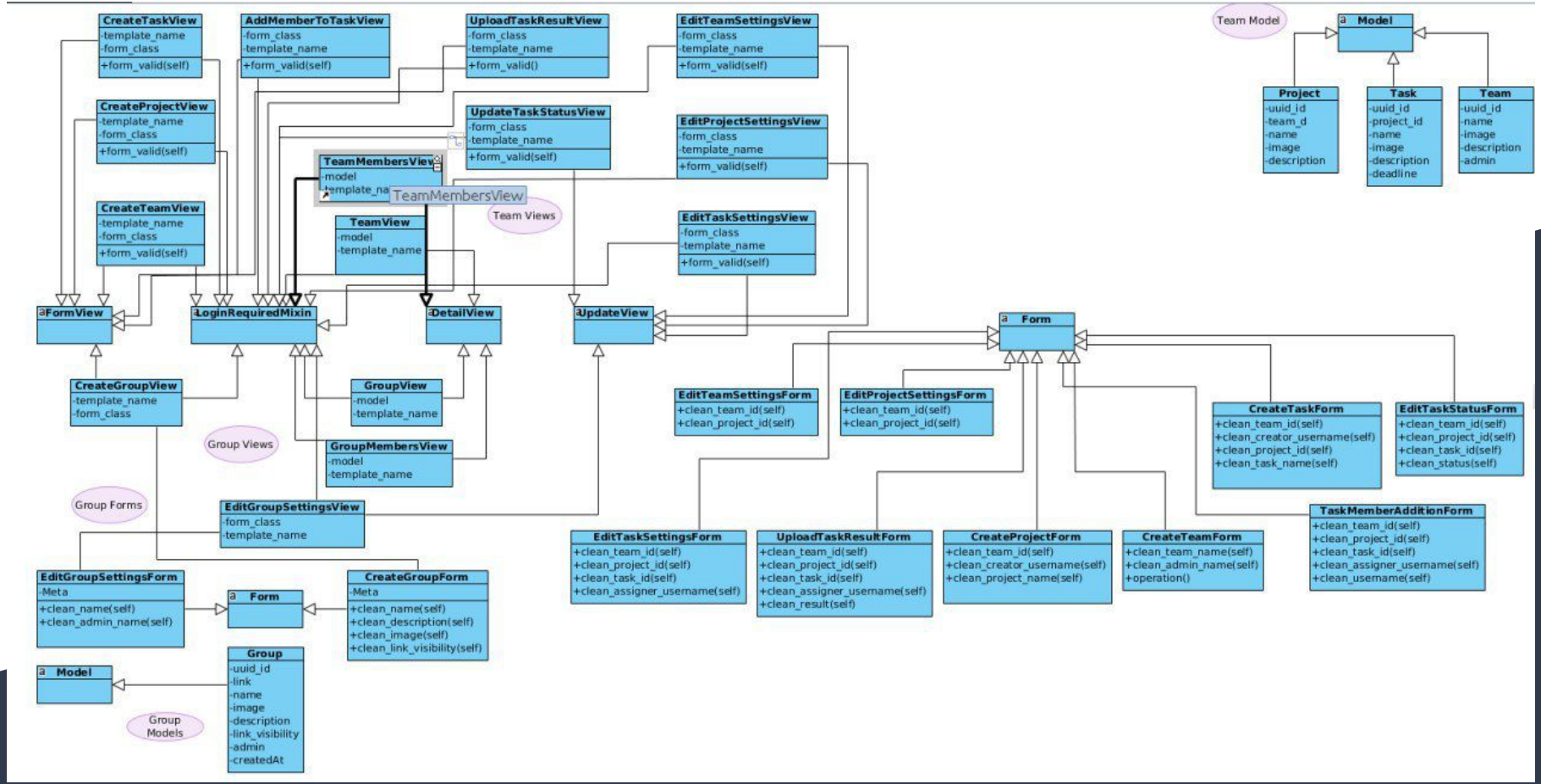


Class Diagram



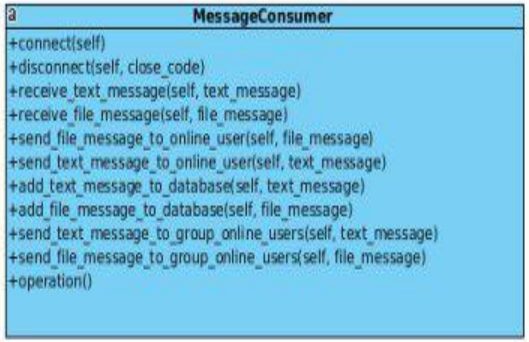


Class diagram

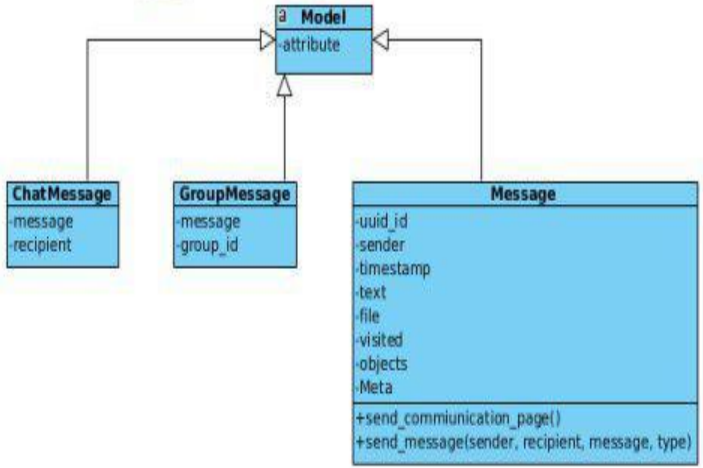


Class diagram

Consumer(websocket)



Models

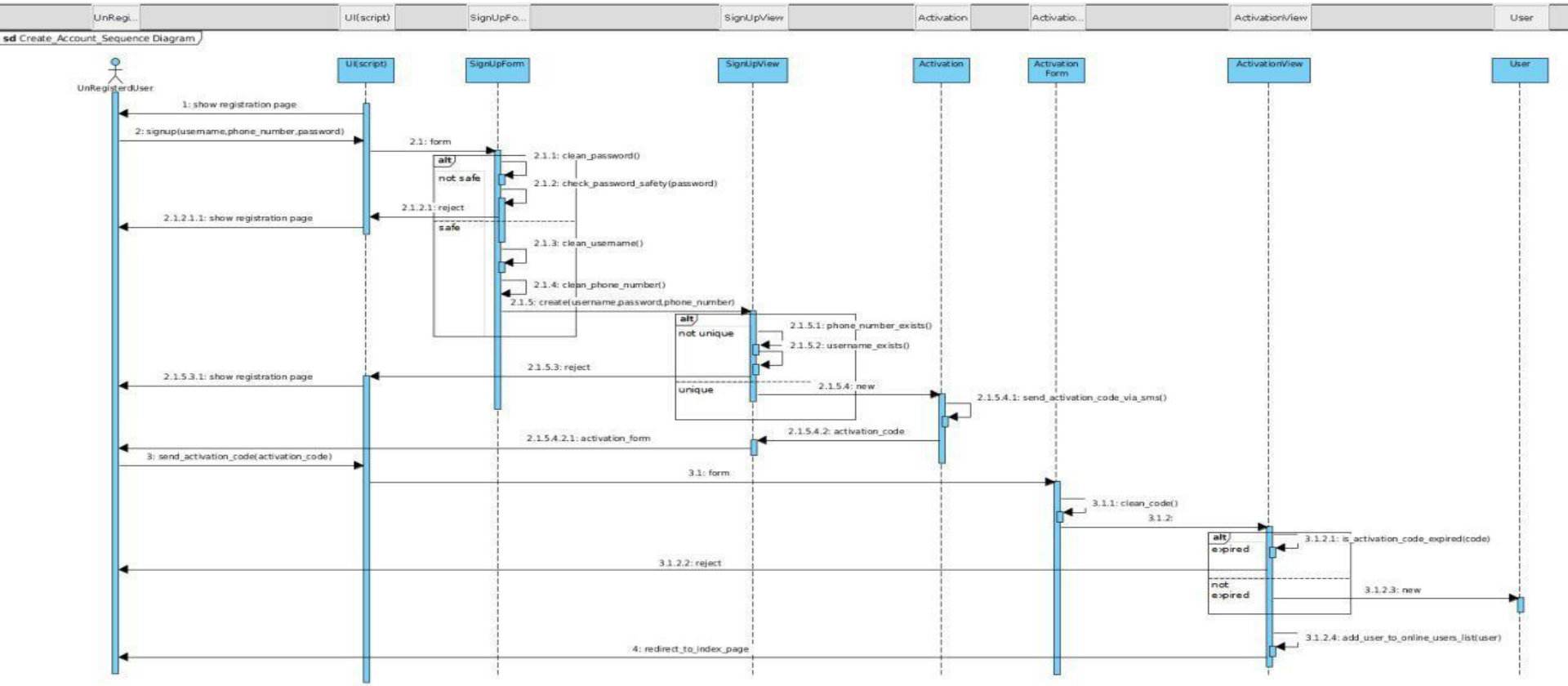


Message View methods

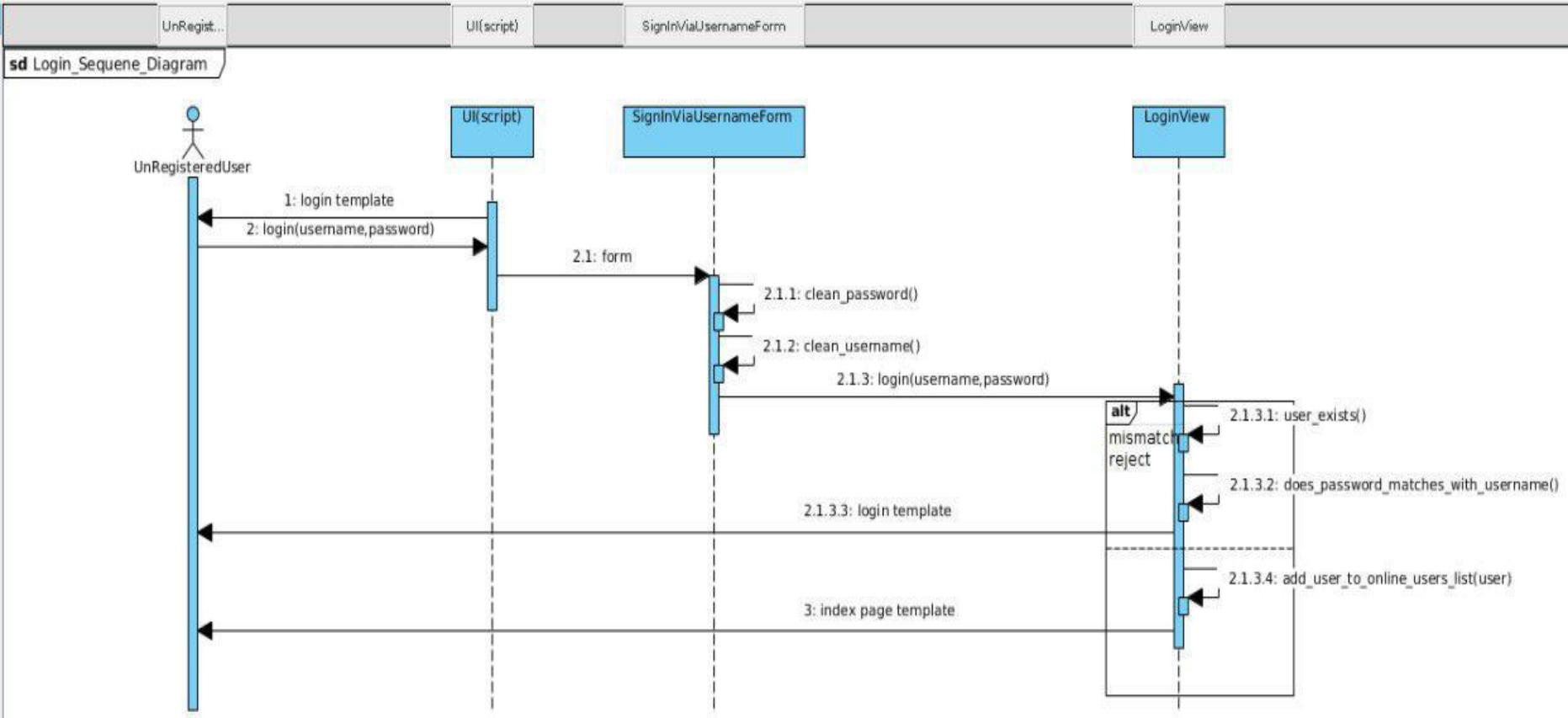
```
@login_required
@ajax_required
def load_commiunication_page_with_user(request):

@login_required
@ajax_required
def load_commiunication_page_in_group(request):

@login_required
@ajax_required
def send_message(request):

@login_required
@ajax_required
def check_recipient_existence(request):
```

sequence diagrams:

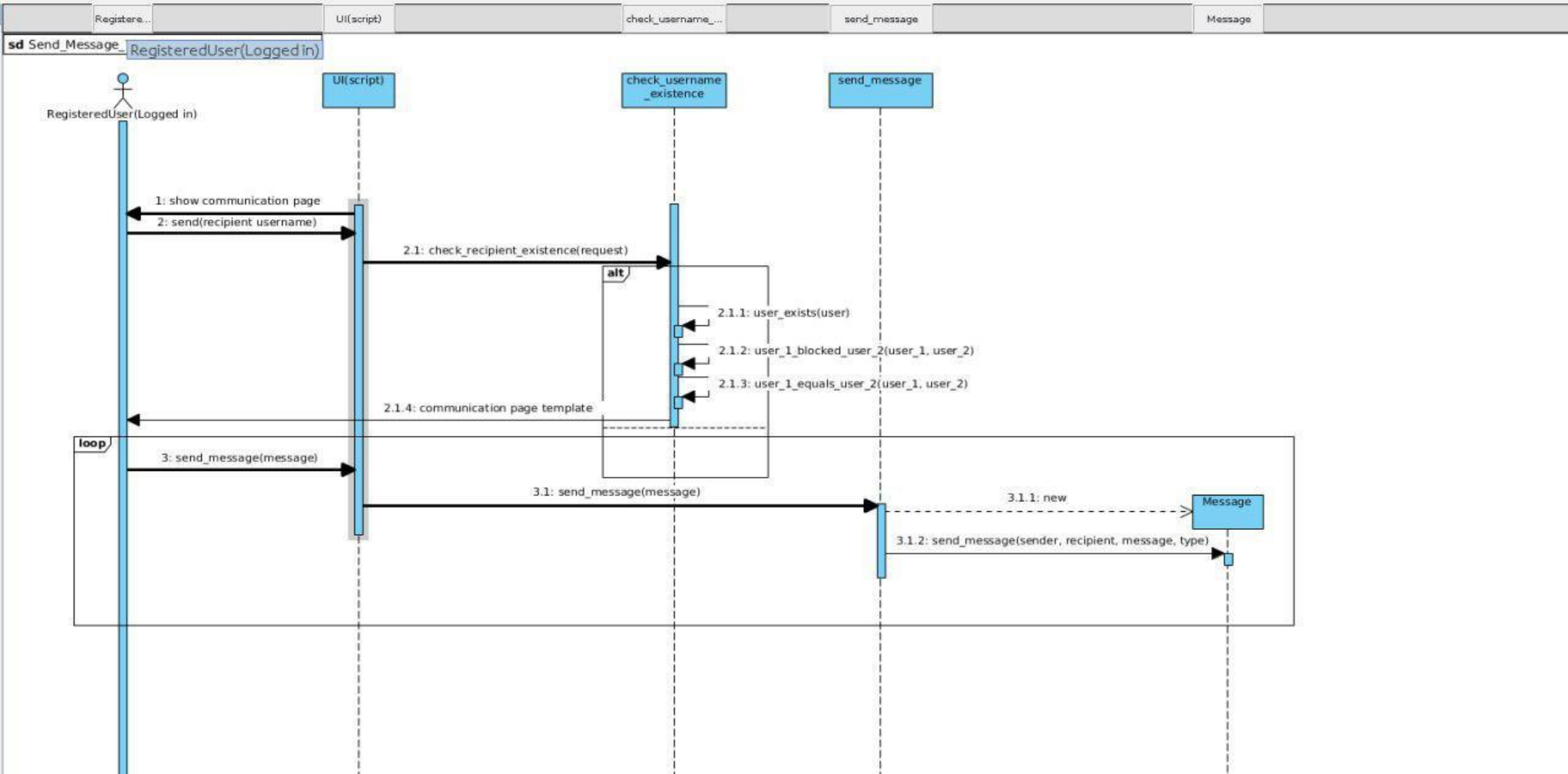
Create Account



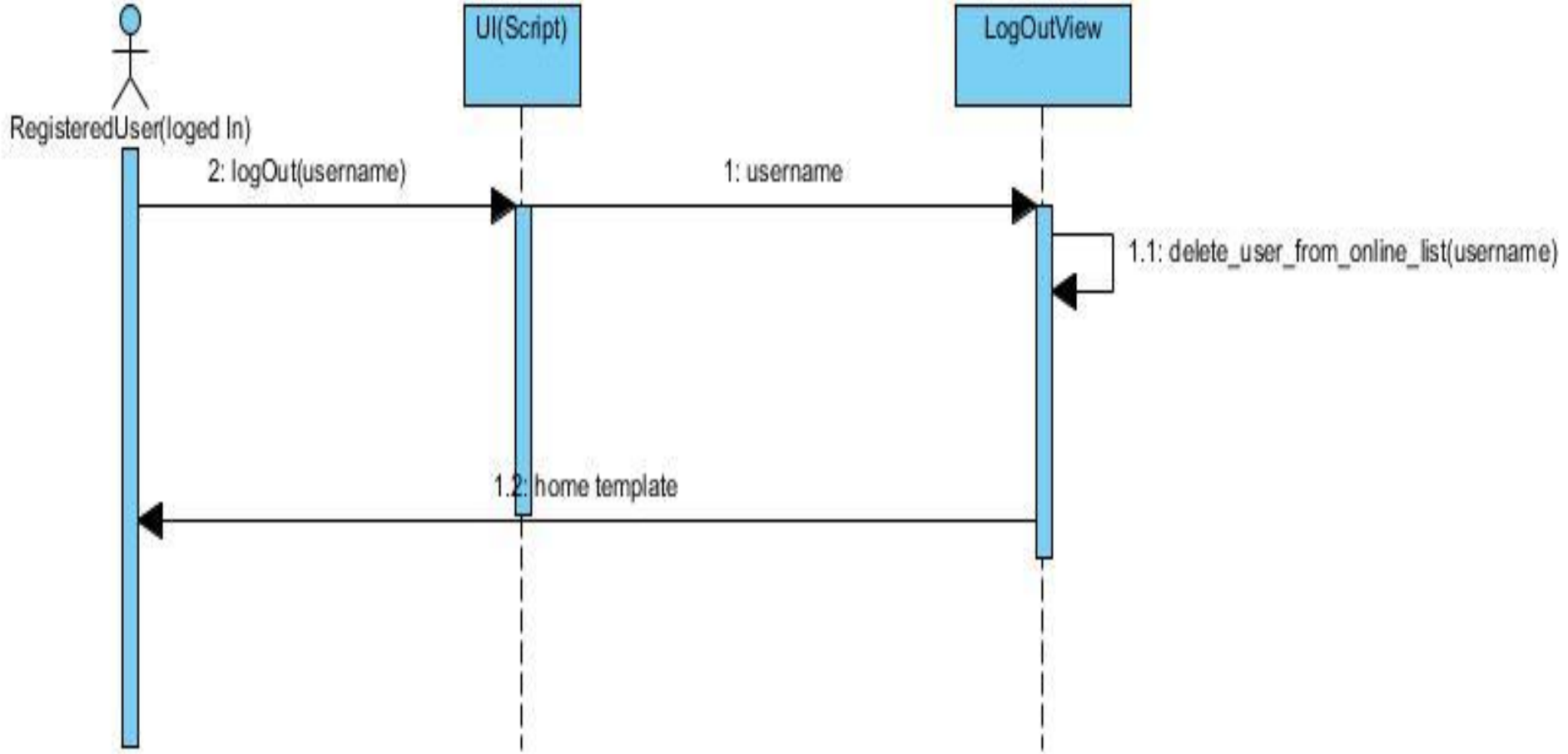
login


UnRegist...
UI(script)
SignInViaUsernameForm
LoginView
sd Login_Sequene_Diagram
UnRegisteredUser
UI(script)
SignInViaUsernameForm
LoginView
1: login template
2: login(username,password)
2.1: form
2.1.1: clean_password()
2.1.2: clean_username()
2.1.3: login(username,password)
alt
mismatch
reject
2.1.3.1: user_exists()
2.1.3.2: does_password_matches_with_username()
2.1.3.3: login template
2.1.3.4: add_user_to_online_users_list(user)
3: index page template

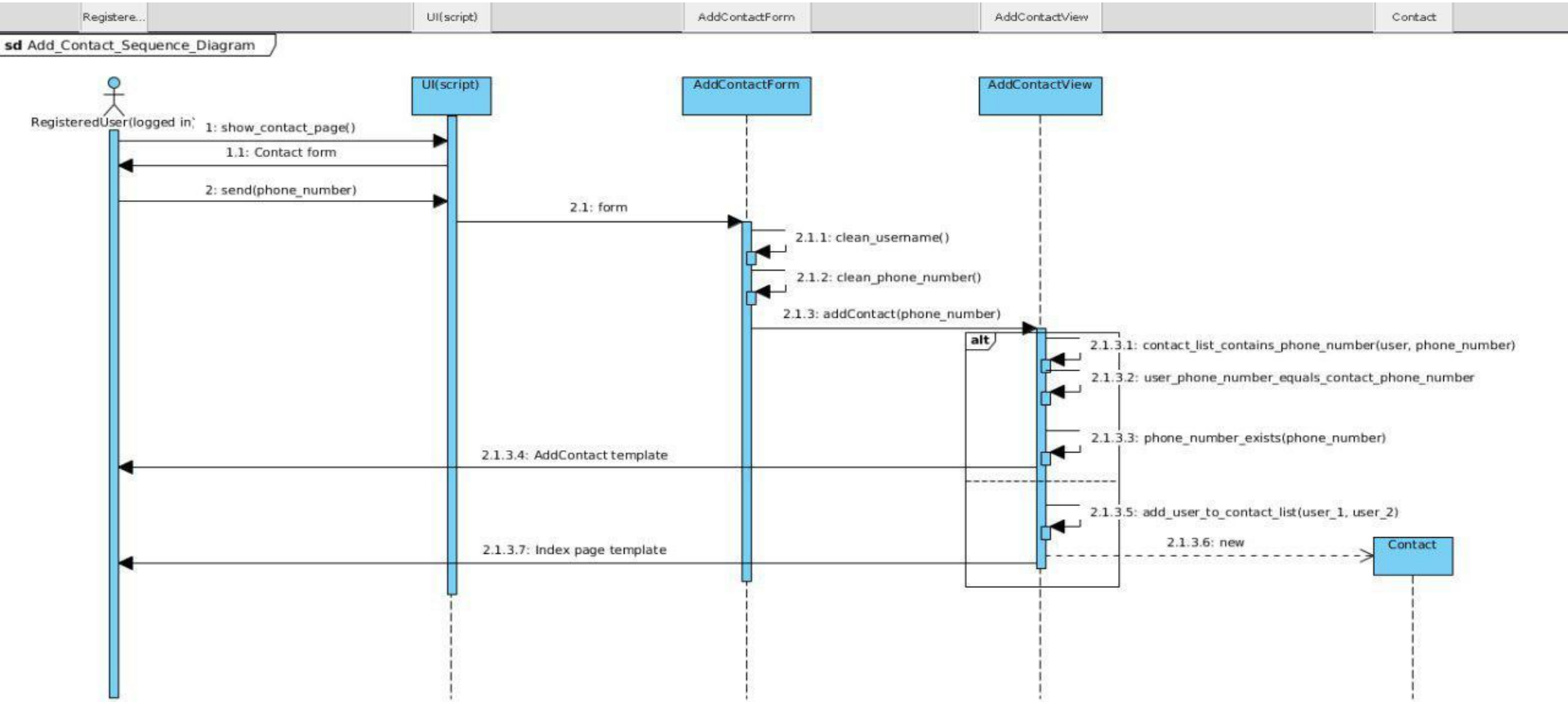
Send Message to user



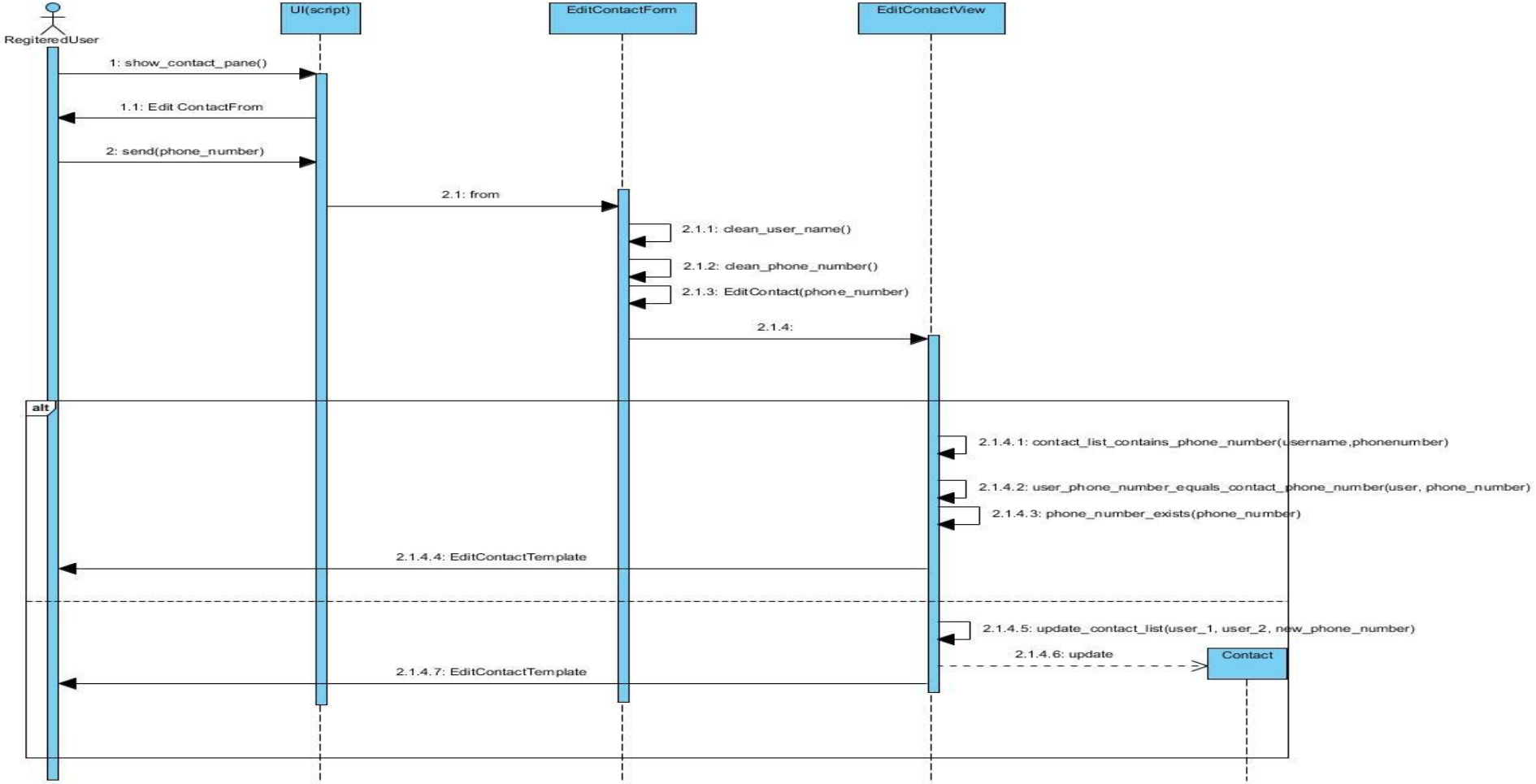
logout


RegisteredUser(loged In)
UI(Script)
LogOutView
2: logOut(username)
1: username
1.1: delete_user_from_online_list(username)
1.2: home template

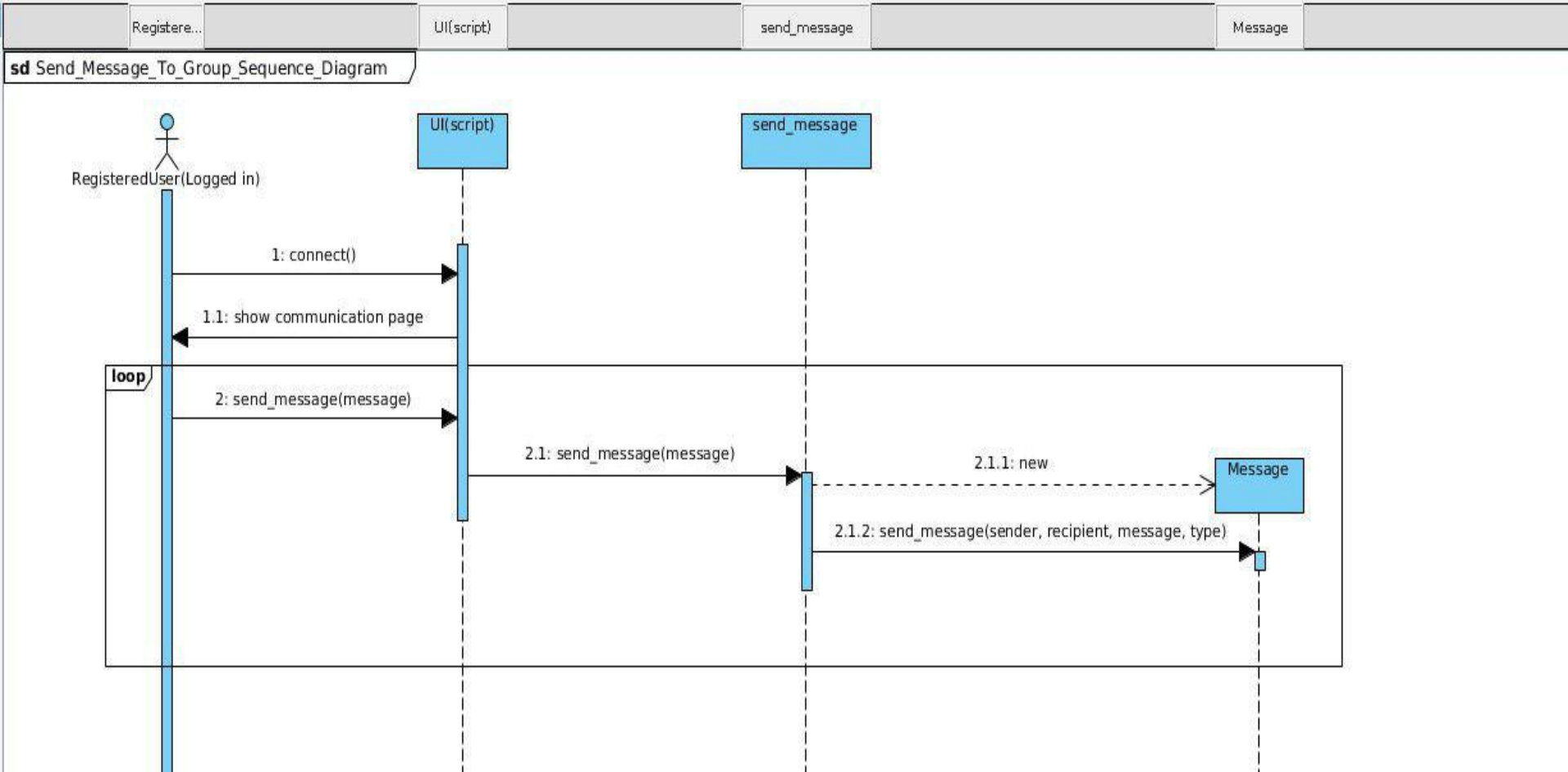
Add Contact


Registere...
UI(script)
AddContactForm
AddContactView
Contact
sd Add_Contact_Sequence_Diagram
RegisteredUser(logged in)
1: show_contact_page()
1.1: Contact form
2: send(phone_number)
2.1: form
2.1.1: clean_username()
2.1.2: clean_phone_number()
2.1.3: addContact(phone_number)
alt
2.1.3.1: contact_list_contains_phone_number(user, phone_number)
2.1.3.2: user_phone_number_equals_contact_phone_number
2.1.3.3: phone_number_exists(phone_number)
2.1.3.4: AddContact template
2.1.3.5: add_user_to_contact_list(user_1, user_2)
2.1.3.6: new
Contact
2.1.3.7: Index page template

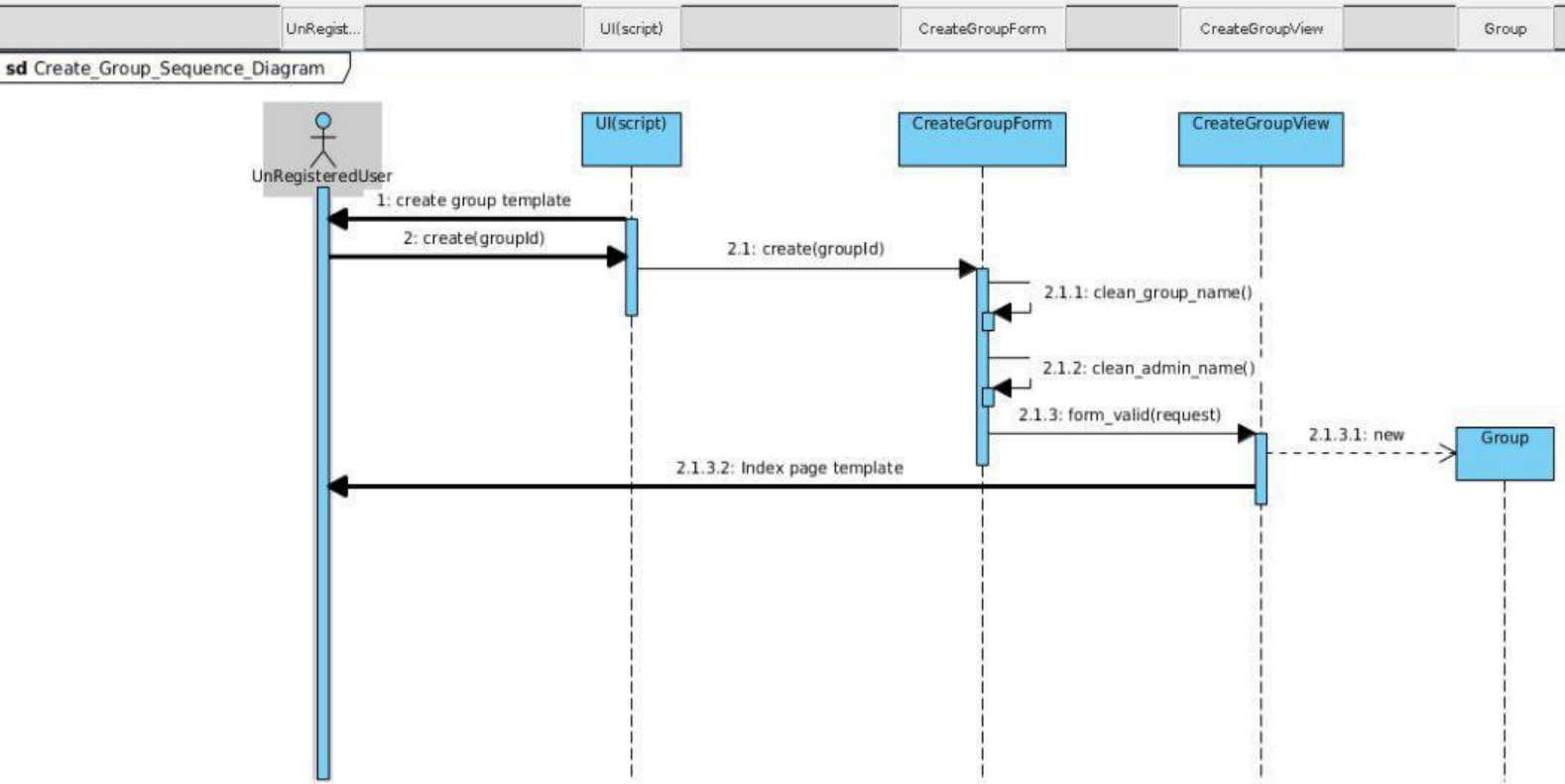
Edit Contact



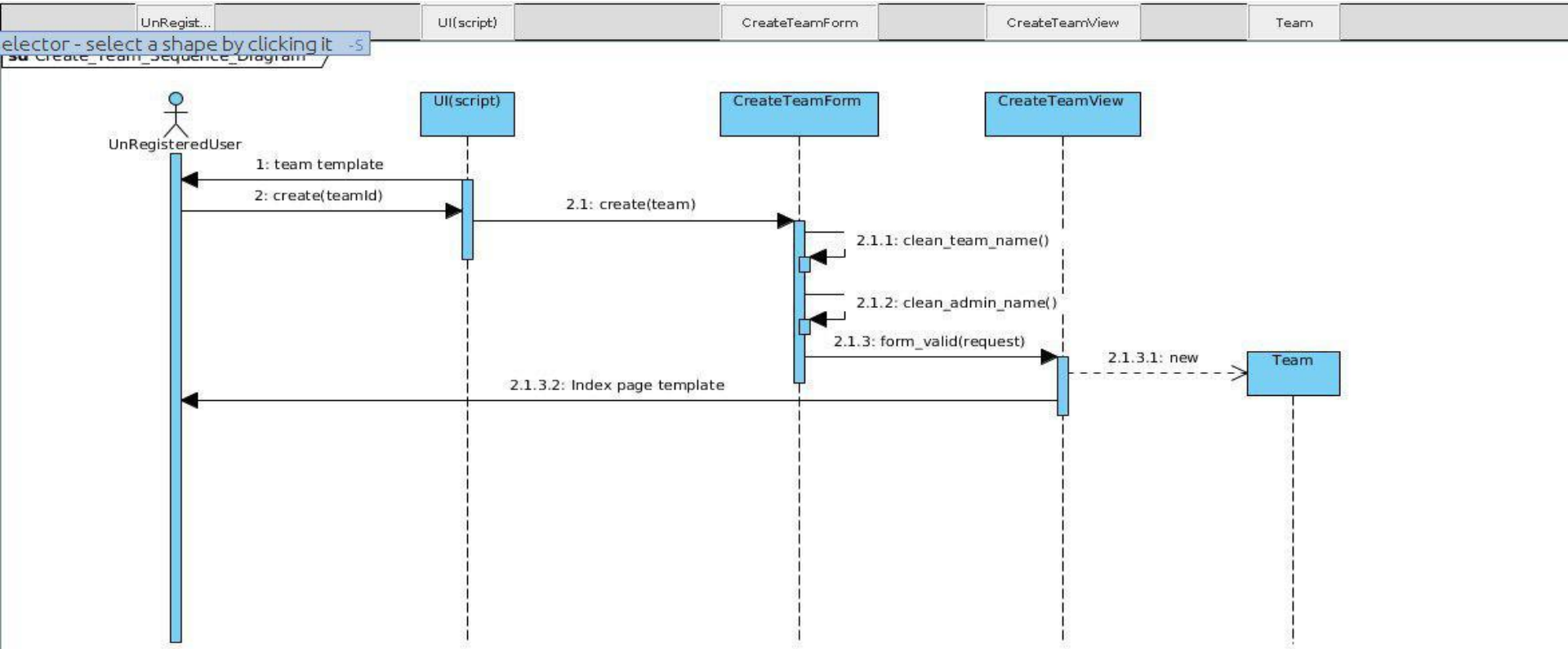
Send Message to group



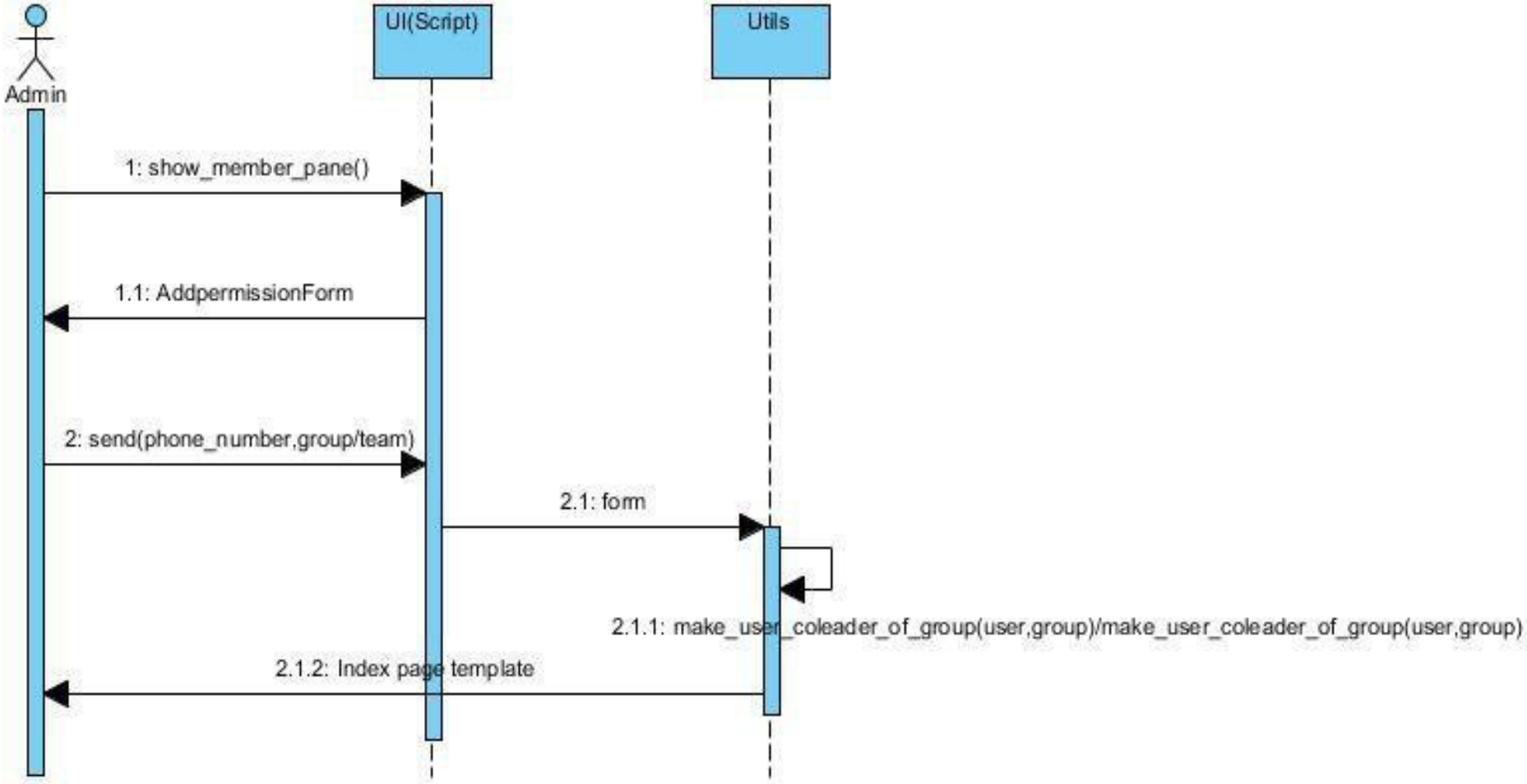
create group



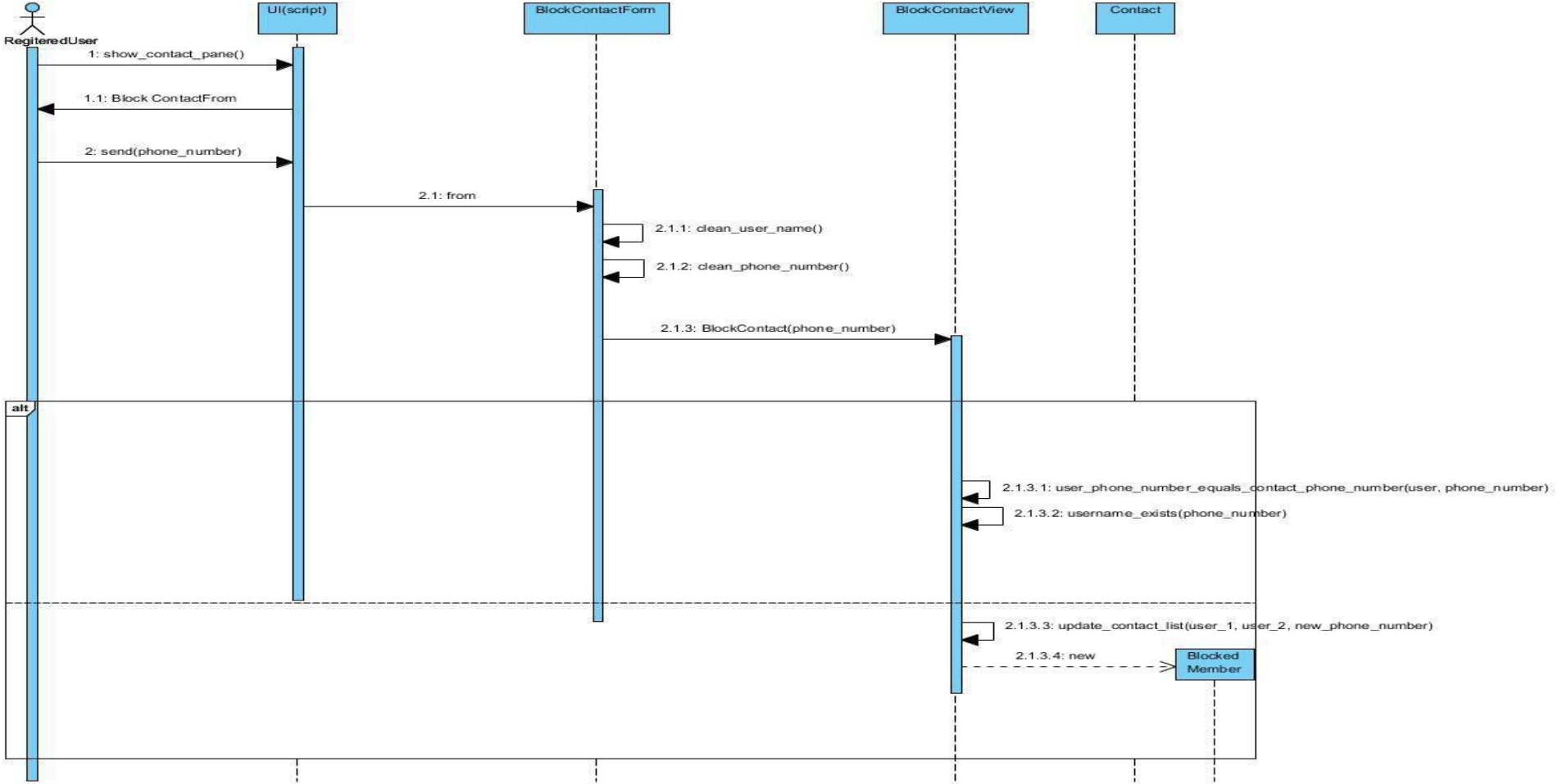
create team


UnRegist...
UI(script)
CreateTeamForm
CreateTeamView
Team
elector - select a shape by clicking it -s
UnRegisteredUser
1: team template
2: create(teamId)
2.1: create(team)
2.1.1: clean_team_name()
2.1.2: clean_admin_name()
2.1.3: form_valid(request)
2.1.3.1: new
2.1.3.2: Index page template

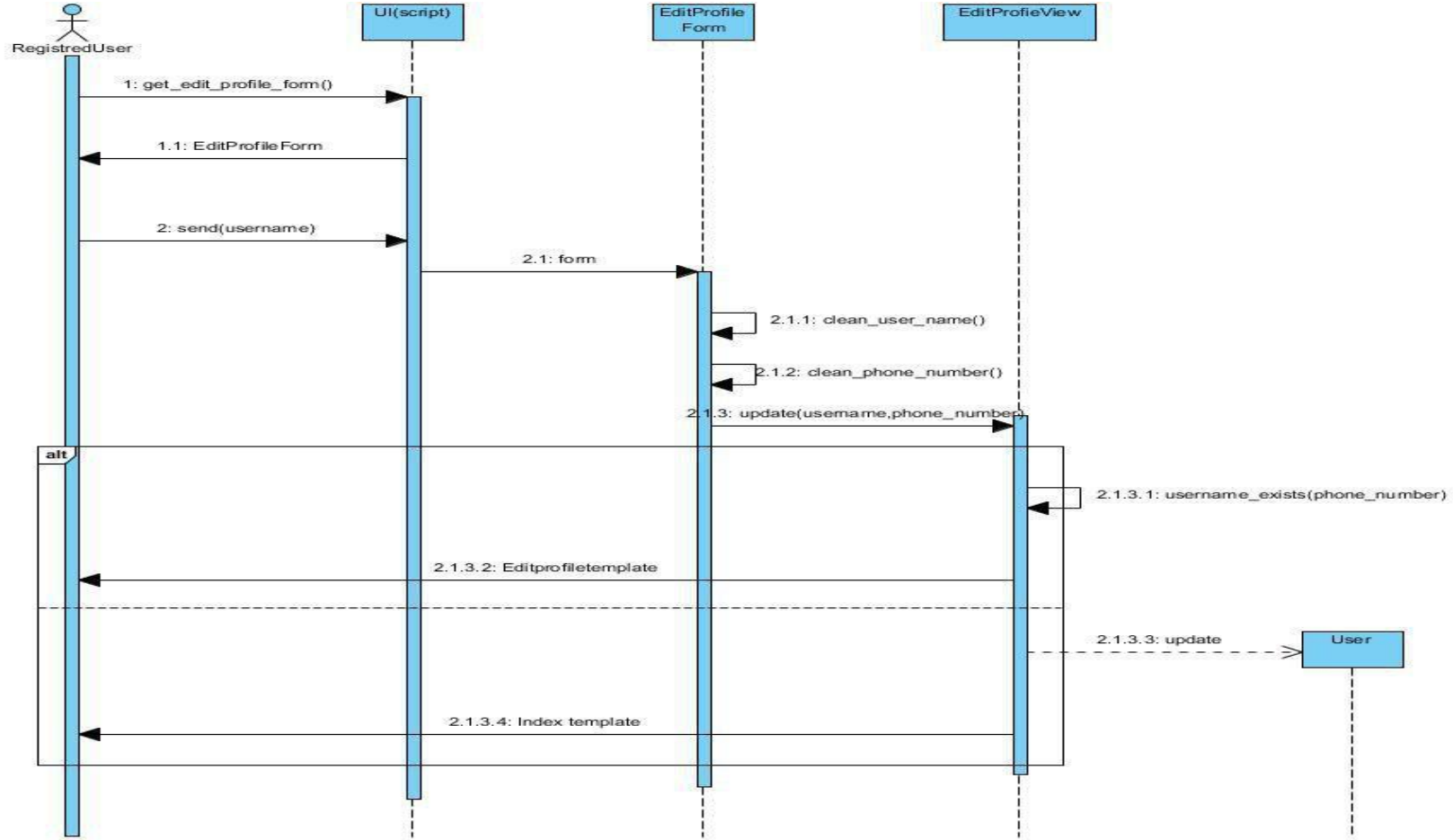
13) add permission



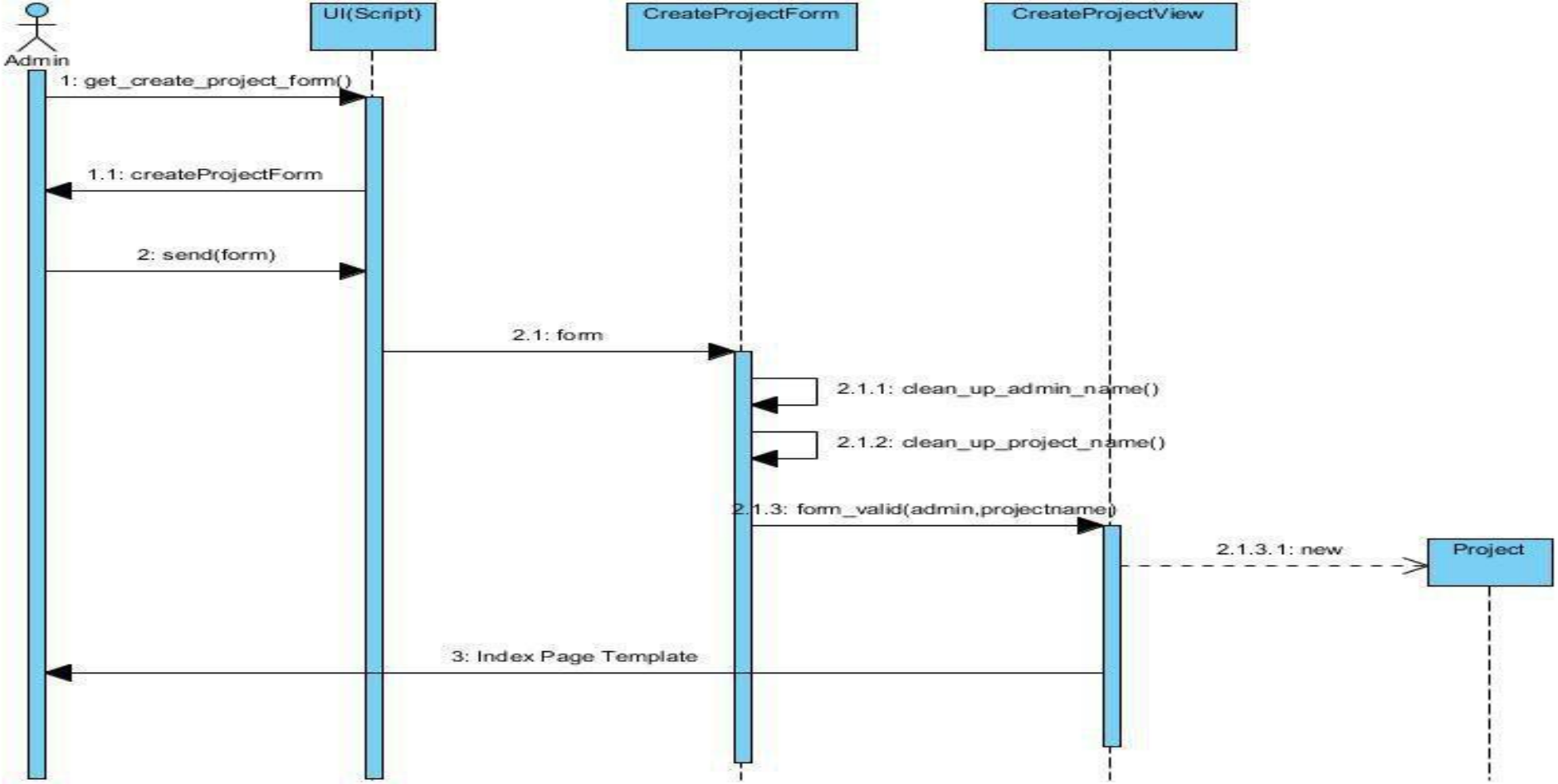
14) block user



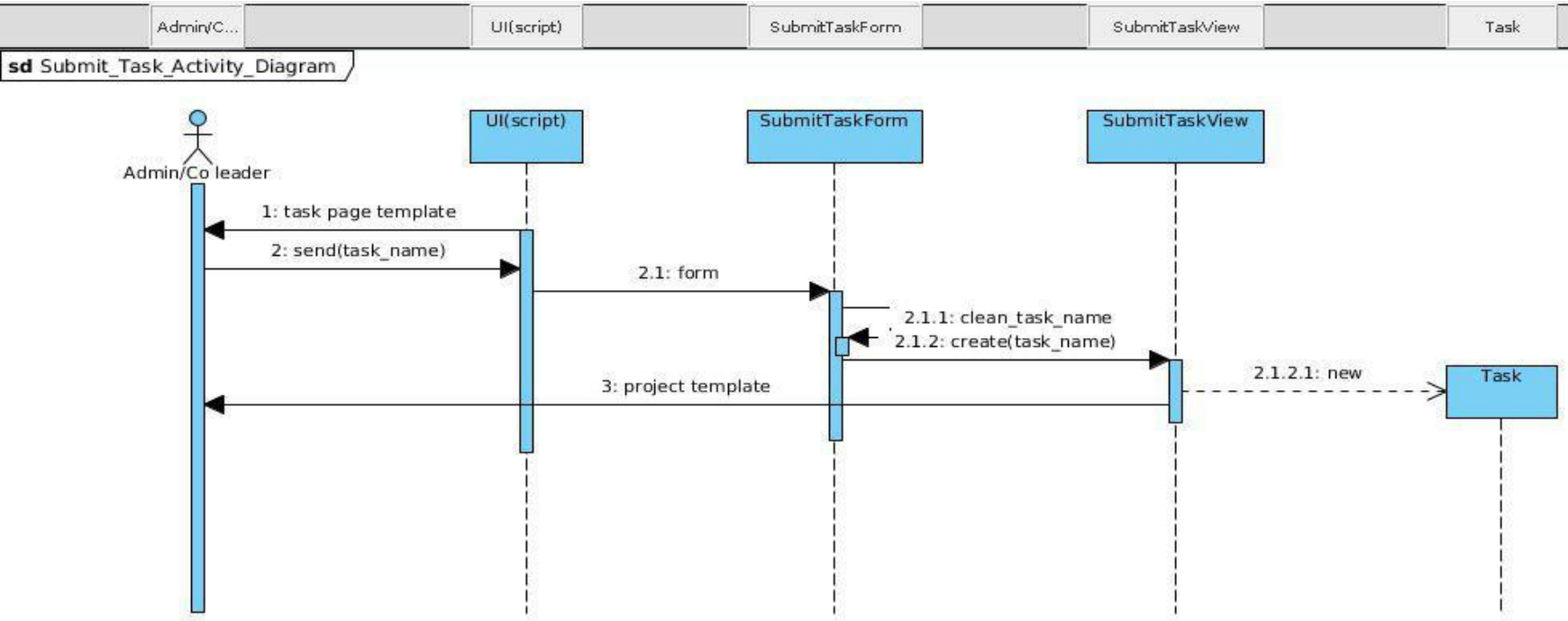
15)edit profile



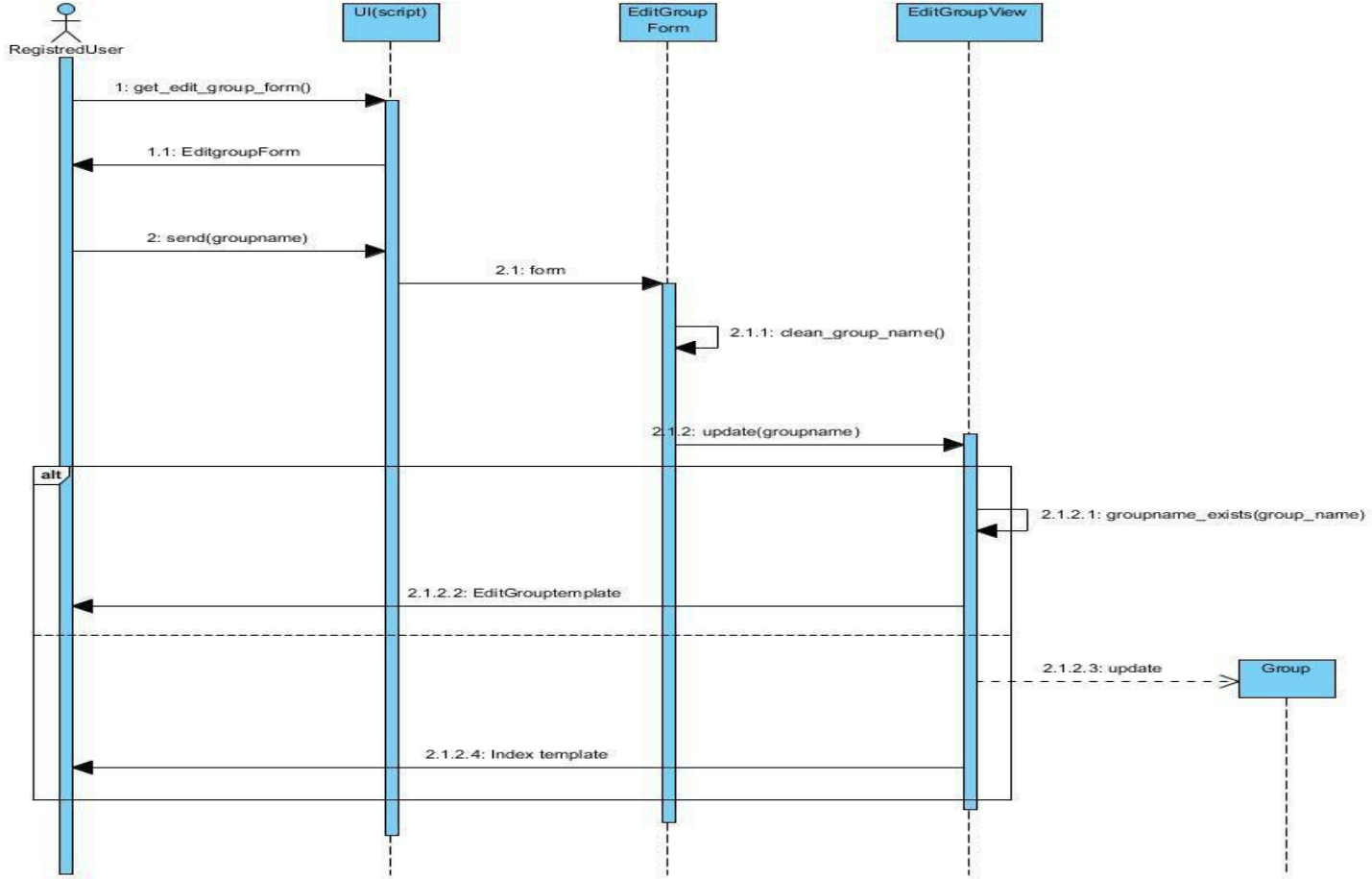
16)create project



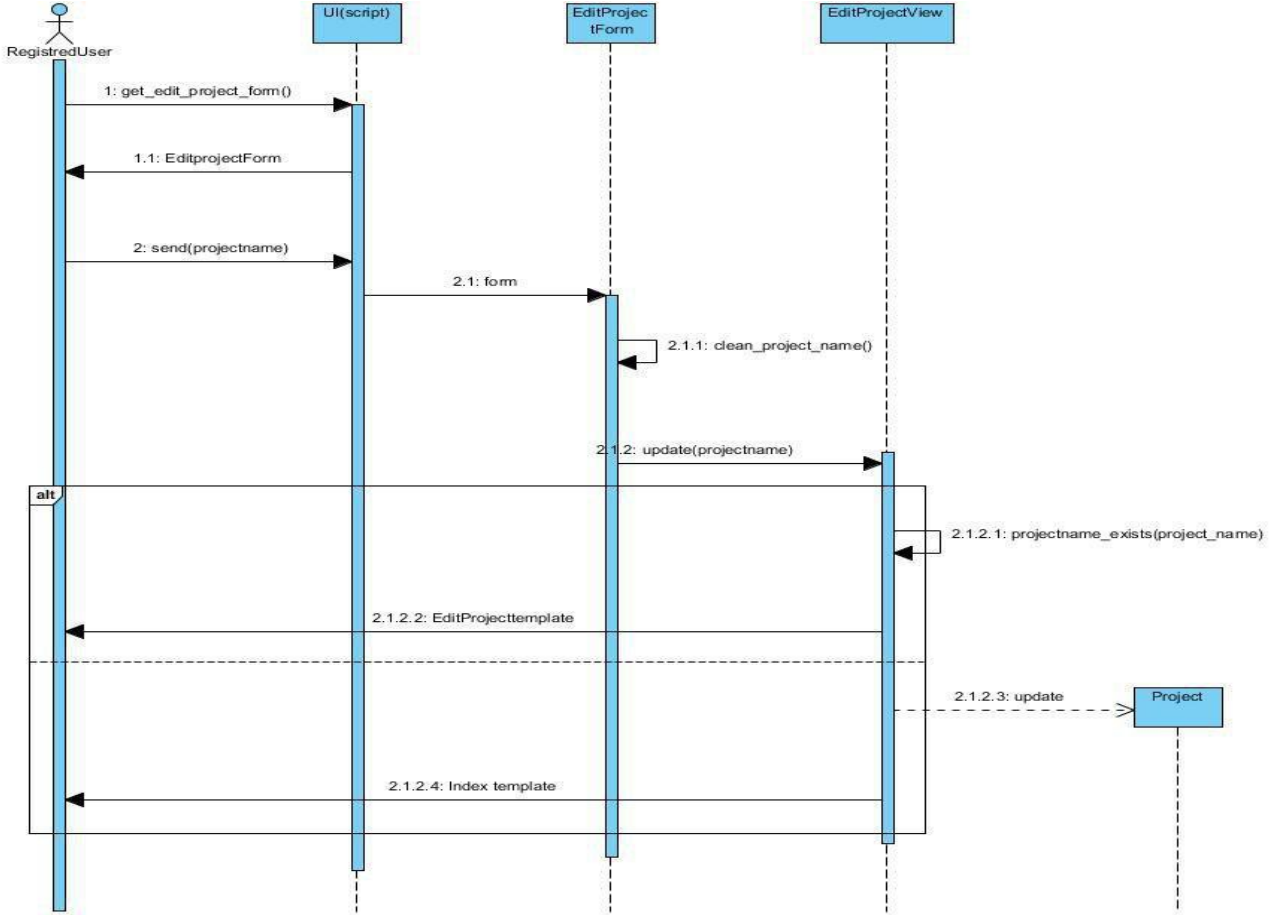
17)submit task



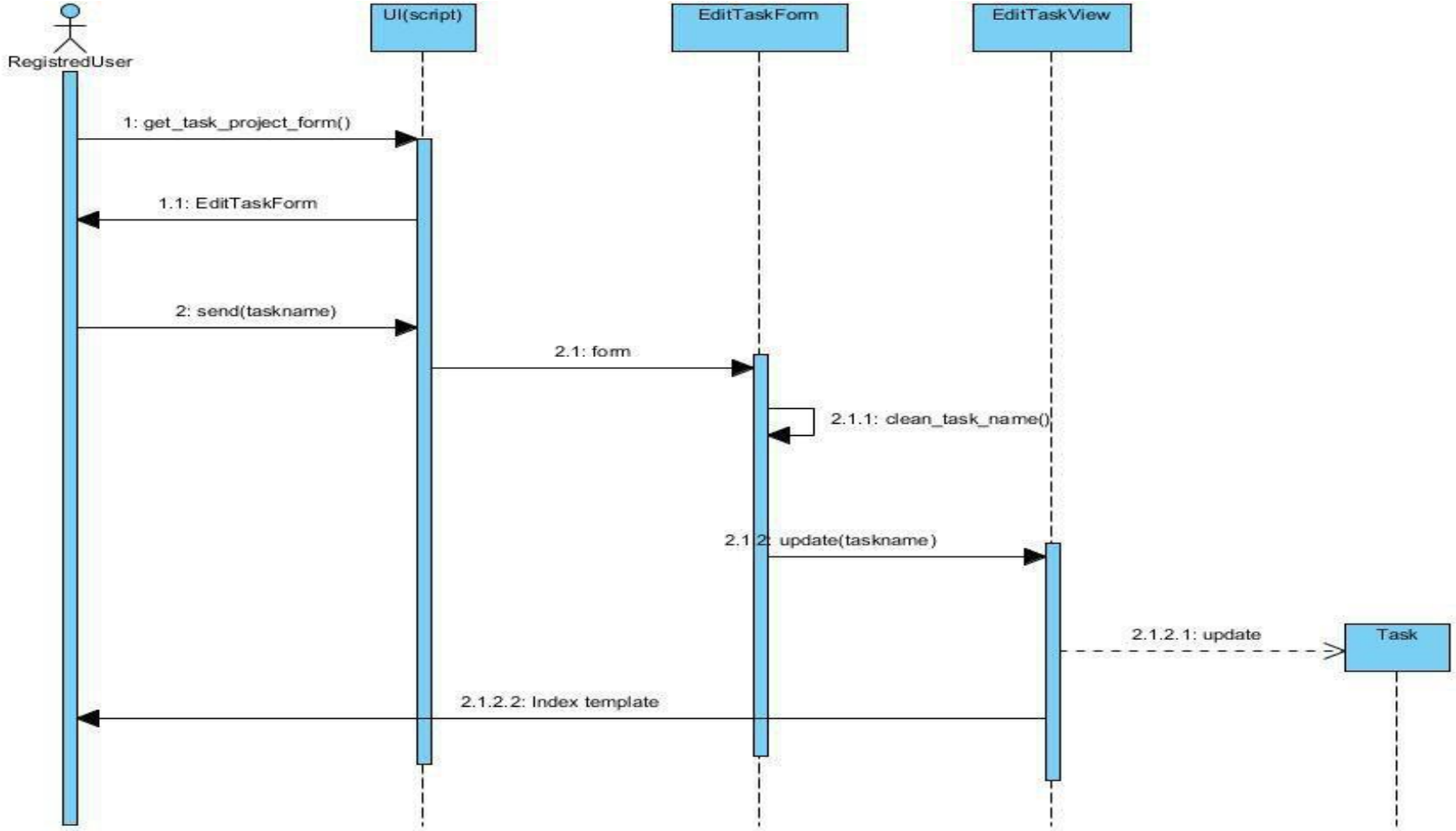
edit group settings



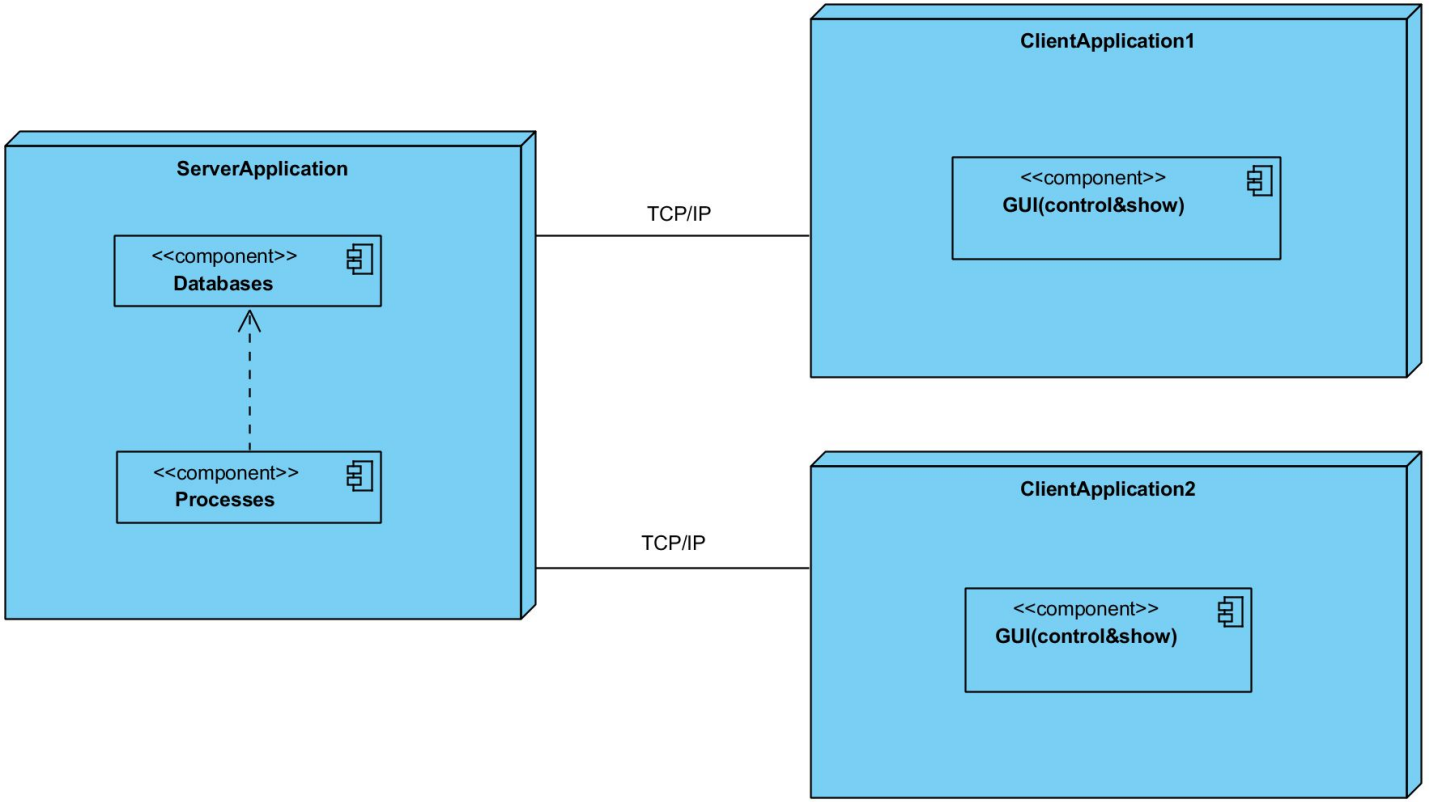
edit project settings



edit task settings



Component & Deployment Diagram




refactoring
meta-programming
declarative
intentional
supporting
model inconsistency
software evolution
model-driven
software
concept
analyse
metrics
formalism
ontology
study
high-level
model
design pattern
detection
unanticipated
graph-based
support
using
software engineering lab
reuse
object-oriented
taxonomy
uml
architectural
future
language-independent
analysing
through
inconsistency management
framework
transformation
graph transformation
evolution
model refactoring
uml model
model transformation
description logic
tool
challenge
formal
approach
comparison
classification
detect
view
umons
consistency maintenance
dependency
software life-cycle
application
simulation
generate
software development
concept analysis
fine-grained
detecting
trend
solution
large-scale
conflict
engineering
system
loading
semantics
behaviour
implementation
interfaced
uniform
state-of-the-art
foundation
mobile
lan
software architecture
automatic
code
program
incremental
analysis

System Evolution

- در نسخه های بعدی نرم افزار امکان ایجاد کانال ارائه می شود مانند گروه با این تفاوت که فقط ادمین می تواند در کانال محتوا به اشتراک بگذارد.

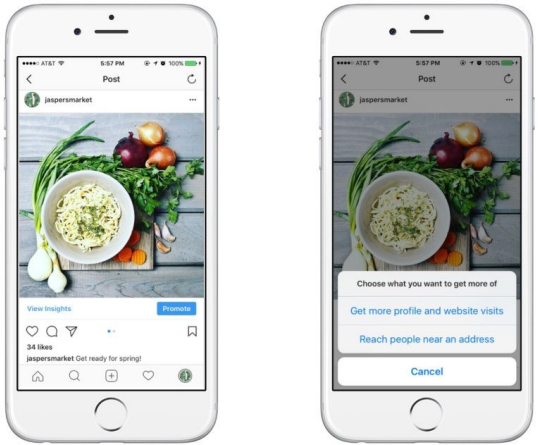
System Evolution

- امکان ویرایش یا حذف پیام ها نیز در نسخه های بعدی سیستم ارائه می شود.
- امکان باز ارسال پیام بدون نقل قول در نسخه های بعدی ارائه می شود.

System Evolution

- امکان داشتن ربات (bot) در نسخه های بعدی ارائه می شود به گونه ای که کاربران می توانند این ربات ها را بسازند و به برنامه اضافه کنند.

System Evolution



- امکان اشتراک گذاری تصویر و ویدیو به عنوان پست و پسندیدن پست ها و ایجاد دیدگاه برای پست ها در نسخه های بعدی سیستم طراحی و ارائه خواهد شد.

System Evolution

- اضافه شدن امکان تماس تصویری در نسخه های آینده ی نرم افزار یکی از گزینه های تکامل سیستم است.

System Evolution

- امکان ارسال موقعیت(Location) در نسخه های بعدی نرم افزار ارائه خواهد شد.

System Evolution

- امکان تنظیم جمله ای به عنوان وضعیت (bio) برای پروفایل کاربر به نرم افزار ارائه خواهد شد.

پروسه ارزیابی نیازمندی ها

Requirement validation process

A. **Requirements reviews** The requirements are analyzed systematically by a team of reviewers who check for errors and inconsistencies.

B. **Prototyping** In this approach to validation, an executable model of the system in question is demonstrated to end-users and customers. They can experiment with
this model to see if it meets their real needs.
We have build a prototype of the system in order validate requirements

C. **Test-case generation** Requirements should be testable. If the tests for the requirements are devised as part of the validation process, this often reveals requirements problems. If a test is difficult or impossible to design, this usually means that the requirements will be difficult to implement and should be reconsidered. Developing tests from the user requirements before any code is written is an integral part of extreme programming.

Process Model

Incremental development

Why???

Our software development process model has to be Incremental because not only there is enough possibility of changes in requirements, we ***know some changes right now*** but we are not considering those in first increment. Furthermore Messenger Application users are ***vast varieties of people*** which we can not figure out all requirements once there has to be increments of the software and also prototypes to get some feedbacks in order to design software using an average of customer preferences and making it customizable for each type of user style.

What that is ?

Incremental development is based on the idea of developing an initial implementation, exposing this to user comment and evolving it through several versions until an adequate system has been developed.

Benefits:

A. The cost of accommodating changing customer requirements is reduced.
B. It is easier to get customer feedback on the development work that has been done.
C. More rapid delivery and deployment of useful software to the customer is possible, even if all of the functionality has not been included

Reuse-oriented software engineering

Reuse-oriented software engineering has the obvious advantage of reducing the amount of software to be developed and so reducing cost and risks. It usually also leads to faster delivery of the software. However, requirements compromises are inevitable and this may lead to a system that does not meet the real needs of users.

But the software package we aim to use **(Django)** is well designed for this purpose.

Benefits Of Reuse-oriented software engineering

Increased dependability

Reused software, which has been tried and tested in working systems, should be more dependable than new software. Its design and implementation faults should have been found and fixed.

Reduced process risk

The cost of existing software is already known, whereas the costs of development are always a matter of judgment. This is an important factor for project management because it reduces the margin of error in project cost estimation. This is particularly true when relatively large software components such as subsystems are reused.

Benefits Of Reuse-oriented software engineering

Effective use of specialists

Instead of doing the same work over and over again, application specialists can develop reusable software that encapsulates their knowledge.

Standards compliance

Some standards, such as user interface standards, can be implemented as a set of reusable components. For example, if menus in a user interface are implemented reusable components, all applications present the same menu formats to users. The use of standard user interfaces improves dependability because users make fewer mistakes when presented with a familiar interface.

Accelerated development

Bringing a system to market as early as possible is often more important than overall development costs. Reusing software can speed up system production because both development and validation time may be reduced.

Reuse-oriented software engineering is almost involved in most of **web based softwares** because they have:

A. **Security**
WAFs may include classes to help implement user authentication (login) and access control to ensure that users can only access permitted functionality in the system.

B. Dynamic web pages
Classes are provided to help you define web page templates
and to populate these dynamically with specific data from the system database.

C. Database support
Frameworks don't usually include a database but rather
assume that a separate database, such as MySQL, will be used. The framework may provide classes that provide an abstract interface to different databases.

D. Session management
Classes to create and manage sessions (a number of interactions with the system by a user) are usually part of a WAF.

E. User interaction
Most web frameworks now provide AJAX support (Holdener, 2008), which allows more interactive web pages to be created.

Plan-driven software process

A **plan-driven software process** can support incremental development and delivery. It is perfectly feasible to allocate requirements and plan the design and development phase as a series of increments.

Why ???

A. We aim to use plan-driven approach because a Messenger system is often a large system and it can not be developed by a co-located team (in contrast to agile methods)

B. Long-lifetime systems requires more design documentation to communicate the original intentions of the system and a Messenger Application is almost that kind of system.

C. Messenger Application is involved in social concepts which can arise cultural issues that may affect the system development This usually requires extensive design documentation, rather than the informal knowledge used in agile processes

D. Our team members do not have same level of skills (even have low level of skills on average) which makes agile methods unsuitable (some have to deal with design and some with programming).

E. Messenger Application is definitely subject to external regulation which is required to produce detailed documentation

(Stakeholders list)

- کاربرانی که از سیستم استفاده می کنند.
- سرمایه گذار سیستم.
- برنامه نویسان ، گروه طراحی ، توسعه و نگهداری سیستم (sysAdmins).